INTELLIGENT CONTROL OF AUTONOMOUS SIX-LEGGED ROBOTS BY NEURAL NETWORKS

Prof. Dr. W. Lechner [1] **Dipl.-Ing. Frank Müller** [2]

Fachhochschule Hannover
University of Applied Sciences and Arts
Computer Science Department
D-30459 Hannover, Postbox 920251, Germany

Abstract: Autonomous mobile six-legged robots are able to demonstrate the potential of intelligent control systems based on recurrent neural networks. The robots evaluate only two forward and two backward looking infrared sensor signals. Fast converging genetic training algorithms are applied to train the robots to move straight in six directions. The robots performed successfully within an obstacle environment and there could be observed a never trained useful interaction between each of the single robots. The paper describes the robot systems and presents the test results. Video clips are downloadable under www.inform.fh-hannover.de/download/lechner.php.

Keywords: Intelligent control, Artificial intelligence, Neural networks, Neural controls, Neural-network models, Robotics, Genetic algorithms

1. INTRODUCTION

The computational intelligence of control systems could be demonstrated by the observation of mobile mini robots within an obstacle environment. Only a few simple movement patterns, like forward, backward and diagonal walking, are trained to the recurrent neural network. The important questions to answer are: How do the mini robots perform within an unknown obstacle environment and to what extent could the robots demonstrate intelligence ? Is there an untrained intelligent interaction between the robots ? Referring to this questions, three sixed-legged mini robot assembly kits with relative simple mechanic parts were bought and then equipped with electronic circuits and a microprocessor for robot control.

[1] werner.lechner@inform.fh-hannover.de
[2] frank.mueller@inform.fh-hannover.de

2. SIX-LEGGED ROBOTS

For nearly ten years autonomous robots with four, six or eight legs were developed. Typical examples are the "Moritz" robot (Zagler et al., 2000), that is able to craw inside a tube system or the "Tarry" robots (Frik et al., 1999), which move within an unknown environment. Compared to these robots, the robots presented in this paper are not build for a specific workload, because the aim of this research project was to demonstrate artificial intelligence of crawling robots.

The used Lynxmotion Hexpod robot kit includes the mechanic parts and twelve servo motors. Servo controllers are additionally required. Each one of the six legs is driven by two servos, the first lifts the leg and the second rotates it. The electronic circuits control the 12 servos, trigger the four infrared emitters, evaluate the signals of two infrared detectors and supply the microprocessor

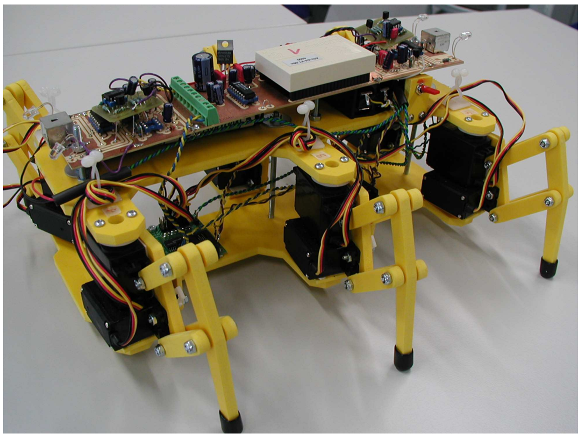

Fig. 1. Picture of the mobile robot

with current. The infrared sensors simulate two eyes for forward looking and further two eyes for backward looking. A picture of the robot is shown in fig. 1. The microprocessor mounted on the top is a "Basic-Tiger" control unit with 1MB RAM, 128kB FLASH-ROM, two serial lines and four A/D-interfaces. The robot is autonomous, but for comfortable testing there are optional interfaces for external power supply and a serial line interface to connect the robot to a computer.

The motion is calculated on the basis of cycling functions with different radials for the left and right legs and increasing or decreasing phase angles. Each one of the legs performs its own elliptic pattern. Fig. 2 sketches the movement principle. While three legs (fl = front left, mr = middle right, bl = back left) contact the ground and carry the weight of the robot, the the other three legs (fr = front right, ml = middle left, br = back right) are lifted and rotated in the meantime. The robot turns left, if the radials of the left legs are smaller than the radials of the right legs. Fig. 3 displays the steering signals of the 12 servo motors in the case of a forward movement. A complete forward step consists out of 12 angle increments each 30 degree wide. For continues movement the sequences of these signals are repeated. Decreasing phase angles lead to backward movements. Symmetric variations of the radials influence the speed of the robot and in the case of different radials on the left and right sided

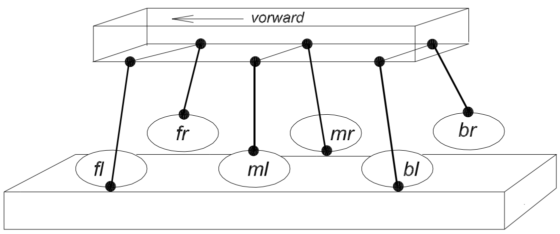

Fig. 2. Movement patterns of the robot legs

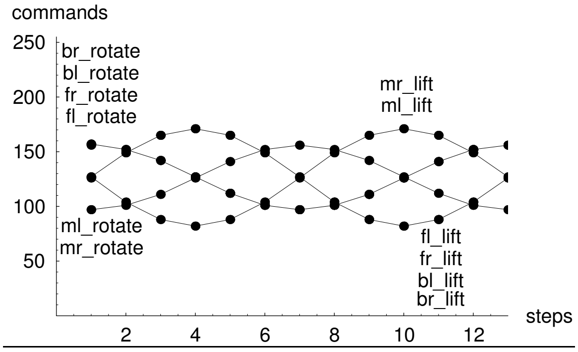

Fig. 3. Steering signals

legs, the robot is able to perform turns. All the movement parameters were optimally adjusted by experimental steering sequences, because the mass of the robot, the mechanic constants and even the limited power consumption influence the dynamic of the movement pattern.

3. NEURAL NETWORK DESIGN

The neural network should be capable to learn the robots basic movement patterns, but in order to demonstrate intelligent control within an environment of arbitrary placed obstacles, it is necessary that the neural network offers dynamic output signals and is able to generalize to a high extend. A further pragmatic condition for this neural network design was the needed relative fast convergence of the training algorithms. Due to all of this reasons a recurrent neural network topology with a limited number of feedback signals was selected (fig. 4). The topology is characterized by a backward linked chain of neurons in the hidden layer. The neurons within the hidden layer are connected to the neurons of the output layer and backward linked to the input of each neighborhood neuron up and below the corresponding neuron. The neuron at the bottom of the hidden layer sends its output signal back to the input of

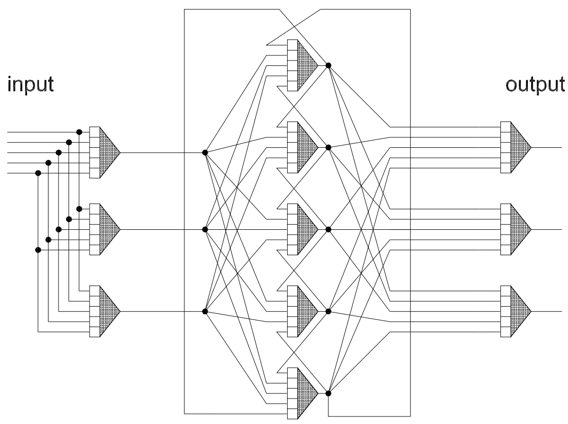

Fig. 4. Recurrent neural network

the neuron on the top of the hidden layer and the same is done in reverse. This topology was first published 1997 by Hotop et al. and could be interpreted as a simplification of the Elman(1990) network.

For the implemented neural network the L=12 neurons of the input layer evaluate S=4 infrared sensor signals and an clock signal C=1. The hidden layer includes K=24 neurons. The L=12 neurons in the output layer control the 12 servos of the robot. Therefore the total number of weights sums up to 684.

$$(S + C) \cdot N + (N + 2) \cdot K + K \cdot L = 684$$

In fig. 4 the parameters are $N = 3; K = 5; L = 3$.

4. GENETIC ALGORITHMS

Pham et al. (1999) demonstrated, that genetic training algorithms could be successfully applied for recurrent neural networks of the Elman typ. Fig. 5 shows the principle of the developed special genetic algorithms, that start with eight matrix sets of neural weights (populations). The algorithm minimizes the quadratic error difference between the network output and the defined and pre-calculated movement patterns of the robots. The eight populations are distributed to the eight main boards of a Siemens HPC-Line distributed memory parallel computer (left part of fig. 7). A limited number of iteration loops are performed on each single board. Then the eight weight matrices are reduced to the first board, where crossover and mutation algorithms are carried out. The best population survives and the worst population is replaced by a randomly selected one. The reference signals are equal to the number of movement patterns, that should be trained and each pattern includes twelve single signals (fig. 3). The parallel programmed algorithms calculated for six

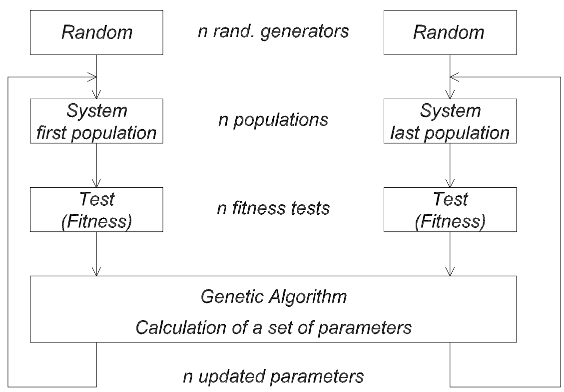

Fig. 5. Genetic Algorithm

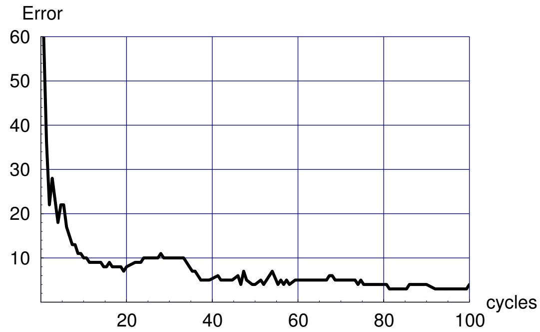

Fig. 6. Error signals during training

movement patterns (6x12=78 output signals) the values of the 684 neural weights within two hours.

Fig. 6 displays the maximal deviation between the output of the neural network and the reference output signals as a function of the number of training cycles. For the HPCLine parallel computer the number of cycles correspond nearly to the amount of minutes of computing time. At the end of the convergent training the neural output signals and the reference patterns show no significant difference. During the training cycles the statistics of the neural weights approximates the Gaussian normal distribution with a vanishing mean value and a standard deviation of ± 0.8.

5. SOFTWARE DEVELOPMENT

The software development system consists of a parallel computer, a standard PC and a programming board for the microprocessor of the robot (fig. 7). First a set of neural weights are calculated on the parallel computer and then downloaded to the PC, which offers software tools to program the microprocessor of the robot and to load the neural weights to its flash memory. Then the robots were observed how they perform within the obstacle environment and based on this test results, the parameters of the neural network could be changed or different movement patterns could be selected and the software development process starts again.

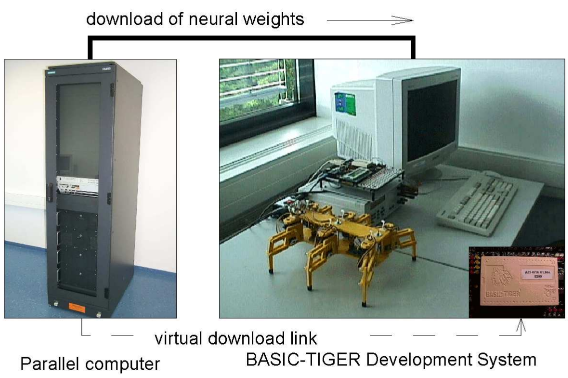

Fig. 7. Software development environment

6. EXPERIMENTS

This section describes several tests of robots, that are trained to move only straight forward or to move in six directions as shown in fig. 8. For example, if the robot detects only front left (fl=1) and a front right (fr=1) obstacles, it performs the trained backward movement. Or, if there is only a back left (bl=1) infrared echo, the robot turns to the right. The four infrared sensors would generally allow a total number of 16 training patterns, however, the below listed six patterns are sufficient for intelligent robot control in an obstacle environment.

(1) $fl = fr = bl = br = 0 \implies$ forward move
(2) $fl = 1 \wedge fr = 1 \wedge bl = br = 0 \implies$ backward move
(3) $fl = 1 \wedge fr = 0 \wedge bl = br = 0 \implies$ turn right
(4) $fl = 0 \wedge fr = 1 \wedge bl = br = 0 \implies$ turn left
(5) $fl = fr = 0 \wedge bl = 1 \wedge br = 0 \implies$ turn left
(6) $fl = fr = 0 \wedge bl = 0 \wedge br = 1 \implies$ turn right

6.1 Single robot with forward movement knowledge

For this test a robot is trained only with the first training pattern(1). In other words, it is said to the robot: If there is nothing to see, move straight forward. If not, make your own choice for a suitable movement. Fig. 9 shows in the left above part the start position of the robot. Moving straight ahead, the robot detects two obstacles with a narrow gap, so the robot is unable to pass through the middle of the obstacles. Surprisingly the robot immediately turns left (above right part of fig. 9). Then further left turns could be observed (bottom left part of fig. 9) and then the robot is going to pass the obstacle on the left side (bottom right part of fig. 9). During this tests, the robot sometimes stops first or starts to move even a single step backward or turns right, but in the end the robot passes the obstacle on the left or on the right side. Please remember: Only the straight forward movement pattern(1) was trained.

In the next experiment the gap is set wider. As shown in fig. 10, the robot approaches the obstacle (left above part of fig. 10) and receives reflected infrared signals. But now - compared to the narrow gab of fig. 9 - the robot corrects its direction first to the right (right above part of fig. 10) and then to the left (left bottom part of fig. 10) and then it passes through the middle of the gap (fig. 10).

In both experiments (fig. 9, fig. 10) the robot was controlled by a neural network, that was able to generalize the trained simple advice(1): Move straight forward, if there is no obstacle to detect.

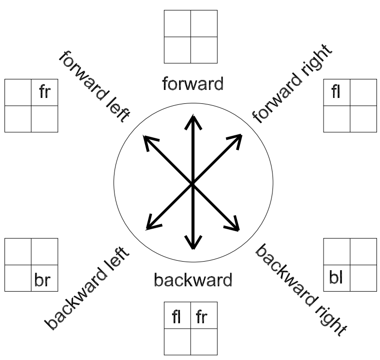

Fig. 8. Training patterns

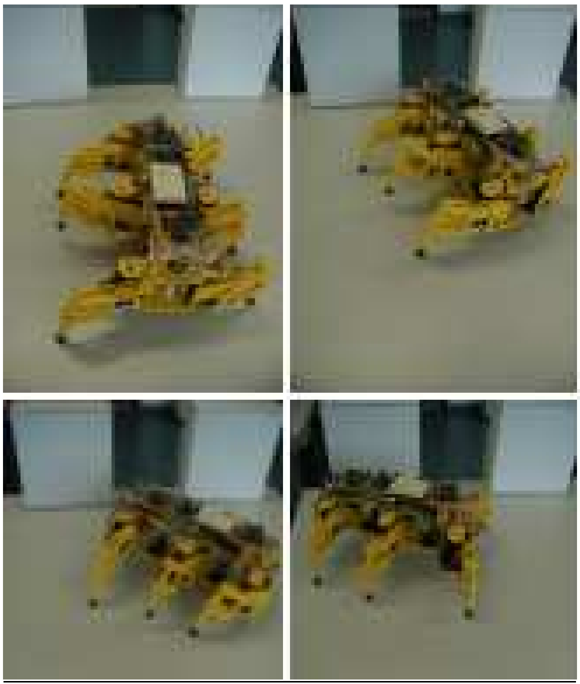

Fig. 9. Forward moving robot approaching a gap that is too narrow to pass through

Fig. 10. Forward moving robot approaching a gap that is wide enough to pass through

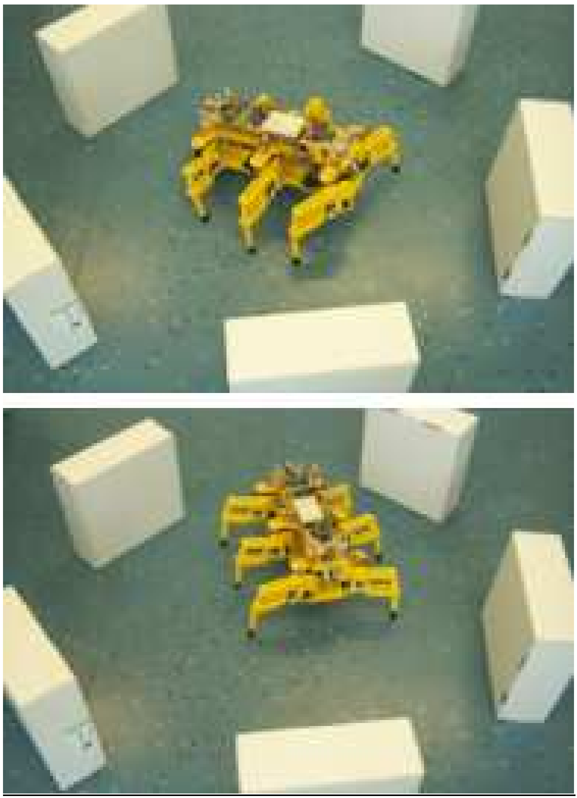

Fig. 11. Escaping robot

6.2 Single robot with six direction knowledge

In the next experiment a robot is trained to move in six 60-degree directions (fig. 8). The above part of fig. 11 shows such a robot that tries to escape through a ring of the obstacles. The robot keeps turning as well as moving backward until the receivers detect a direction that is free of obstacles and then the robots starts to walk in this direction (bottom part fig. of fig. 11) . The robots of fig. 9 and fig. 10 were mostly not able to escape, because the generalization of the single forward movement pattern(1) of this robots was not sufficient for intelligent control within a ring of obstacles.

6.3 Two robots with six direction knowledge

During this experiment two robots with six direction knowledge are used. Fig. 12 shows in the left above part the robots moving towards each other on a collision approach. The robot below detects first the opposite second robot and turns to the right (right above part of fig. 12). The second robot notices infrared reflections on both of its front receivers and immediately stops, although a stop was never trained. During this stop the first robot could pass. In the bottom part of fig. 12 both robots freely pass each other.

The intelligent control of the robots was influenced by an unknown mixture of infrared signals

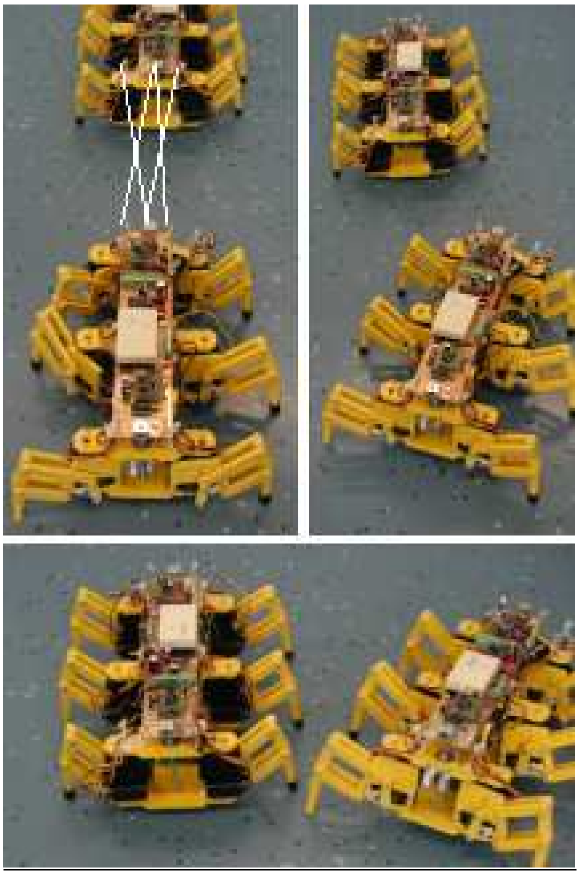

Fig. 12. Collision approach of two robots

(see the manually painted white lines in the above left part of fig. 12). The emitted infrared signals of the first robot are detected by the receivers of the second robot and the same in reverse, because the electronic circuits make no difference between the received signals. The reason for this electronic design was, that the movement of the robots in the case of randomly received infrared signals should be observed.

In every experiment, the robots avoided the collision. Even if two robots walking in a line (left part of fig. 13), the faster robot behind recognizes the first robot, turns immediately to right (right part of fig. 13) and moves away.

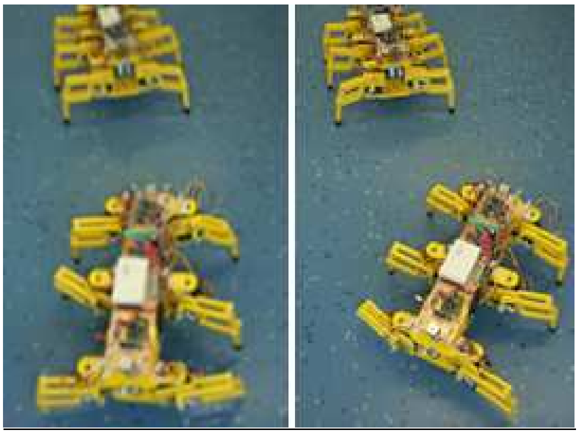

Fig. 13. Two robots walking in a line

6.4 Three robots with six direction knowledge

During this experiments three robots are observed, how they perform within an obstacle environment. Look with a left to right and a top to bottom reading sequence at fig. 14. Starting from a parallel position, all robots detect obstacles, but only the left robot moves forward and escapes through the front gap, while the two other robots perform turning and waiting patterns. While the first robot is heading the obstacle in the second line, the next robot detects the free passage through the first gap and follows the first robot. If the last robot does not see an obstacle, it starts to pass through the obstacles.

The principle of the movements is based on the fact, that any robot walking behind or in front of another robot considers this robot as an obstacle and tries to avoid collision by turning, waiting or walking backward. The robots perform collision-free movements and at least all pass through obstacles (right bottom part of fig. 14).

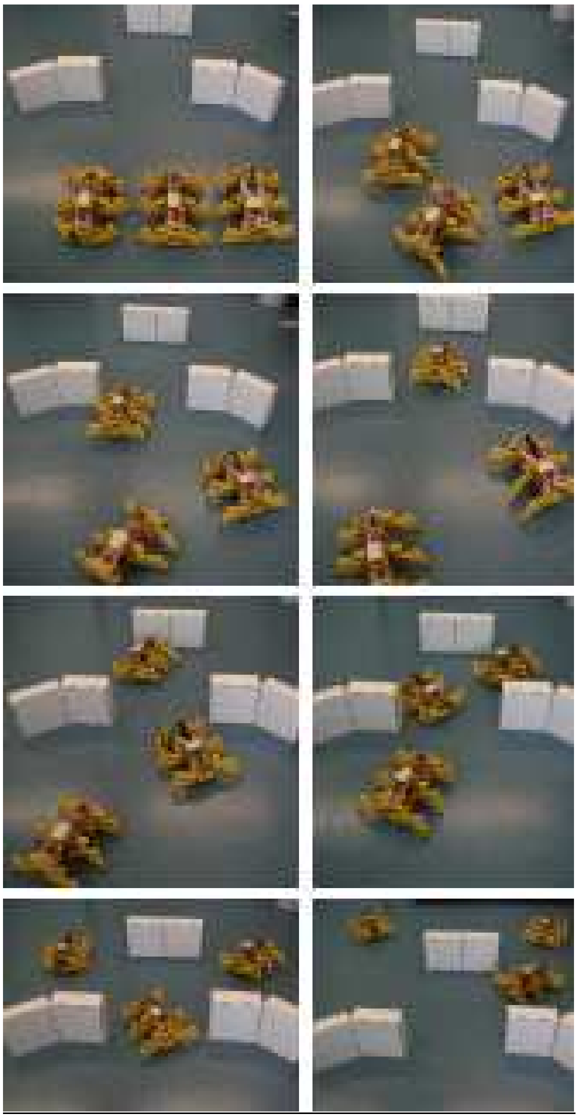

Fig. 14. Three robots passing through obstacles

7. CONCLUSIONS

The paper described the experiments with recurrent neural network driven autonomous robots in order to demonstrate the possibilities of intelligent control. Based on the generalization of straight line movement patterns the robots avoided any collision among each other as well as with obstacles. Fascinating, complicated and never expected movement patterns could be observed during the experiments. The corresponding video clips could be downloaded under www.inform.fh-hannover.de/download/lechner.php.

Due to the four infrared emitters only six logical true/false conditions (fig. 8) were trained. However, during the experiments the receivers detect a signal mixture reflected by an obstacle or direct emitted by another robot in the range $0 \leq x \leq 1$ with 10 increments. Therefore the neural network evaluates 10^4 different input signal sets, each set consists of four real numbers in the above defined range. In other words: Only $6/10^4$ or 0.06% of the neural network input signals are used by the training algorithms.

Lots of experiments with different types and numbers of training patterns demonstrated, that the presented specific recurrent neural network topology with the ring of backward linked neurons in the hidden layer, has the potential for intelligent control. The amount of training patterns was kept very low and the robots performed generally much better than trained.

REFERENCES

Elman J.L. (1990). Finding structure in time, *Cognitive Science*, vol. 14, pp. 179-211.

Frik M., Guddat M., Karatas M. and Losch D. (1999). A novel approach to autonomous control of walking machines. *Proc. of the 2nd Int. Conf. on Climbing and Walking Robots CLAWAR 99*, Portsmouth, UK. Professional Engineering Publishing Limited, Bury St. Edmunds, pp. 333-342.

Hotop H.J. and Lechner W. (1997). In-Flight Wind Speed and Air Temperature Prediction for Commercial Aircraft using Recurrent Neural Networks, *International Conference on Engineering Applications of Neural Networks, EANN 97*, Stockholm, pp. 219-226.

Pham D.T. and Karabogsa D. (1999). Training Elman and Jordan networks for system identification using genetic algorithms, *Artificial Intelligence in Engineering 13*, pp. 107-117.

Zagler A. and Pfeiffer F. (2000). Refined control for the tube crawling robot, *in Climbing and Walking Robots, CLAWAR 2000*, pp. 339-346.