Evolving Neurocontrollers in the RoboCup Domain

Keyan Zahedi, Martin Hülse, Frank Pasemann, Fraunhofer Institut AIS, Sankt Augustin

1. Introduction

Robot intelligence is often associated with the concept of autonomous systems which have to decide and act without central control, external technical guidances, or human assistance. Especially autonomous mobile robots are nowadays conceived of as robots that can operate in complex, dynamically non-trivial environments. They are supposed to be equipped with several types of sensors and with various actuators to solve a desired task. Sometimes they are assumed to develop also communication skills and some kind of social behaviour which allows cooperative interaction – possibly with humans.

To compare the achievements of various approaches in this highly active field the RoboCup competition [1] delivered an exquisite test platform for the behaviour control of autonomous mobile robots. It offers a well-defined environmental setup and is supported by a large and growing community. The global benchmark is to win the RoboCup competition, but it also allows the definition of subtasks on an individual as well as on cooperative level.

A successful neural control of autonomous mobile robots, mimicking a kind of cognitive behaviour, would be interesting to study: Like their biological counterparts, these neural networks are assumed to use recurrent connectivity and dynamical features to generate a goal oriented behaviour. Driven by their sensor inputs, these artificial neural systems will allow to study the appearance and disappearance of their dynamical attractors, i.e. the delicate balance of stability and instability of neurodynamics, features which seem to be crucial for many of the adaptive and higher-information processing capabilities of biological systems. Eventually, this studies will lead to alternative design methods for robust neural controllers avoiding the standard (and often not very efficient) learning algorithms [7].

Being open to the question how a successful neurocontroller may look like, it seems to be appropriate to use an Artificial Life approach to Evolutionary Robotics [6], [8]. This approach was proven to derive interesting solutions to control problems, even without specifying beforehand the number of neurons or the connectivity of the desired networks, [10], [9].

In this contribution we will apply techniques of evolutionary robotics to develop the control for the goalkeeper of the GMD-Musashi RoboCup-team. We consider the goalkeeper as one of the key players in nowadays RoboCup competitions. For some reason, defending the goal seems to be a critical task, as there are almost no convincing goalkeepers presented yet. Most of them tend to defend very poorly or even leave the goal.

Watching real RoboCup matches, one can define two primary goals for the evolution of a neurocontroller. First of all the goalkeeper has to defend the goal from any approaching ball. Defending the goal does not include the sidelines or any other area on the play field. Therefore the primary restriction for the goalkeeper is that the penalty area is not to be left at any time. This paper presents an evolved neural network for behaviour control of a goalkeeper robot, which is structurally minimal and optimal with respect to the described task. It represents an understandable goalkeeper strategy, and selects only a reduced set of sensor informations for its control.

2. General Setup

We use an algorithm, called ENS^3 ("evolution of neural systems by stochastic synthesis") in [9], which evolves neural networks of general recurrent type. It is used mainly for structure development, but it also optimises parameter values like synaptic weights at the same time. Only the number of input and output neurons of a controller is fixed according to a given sensor-motor configuration. Therefore resulting networks can have any number of internal neurons and any kind of connectivity structure, including feedback-loops and self-connections. During the evolution of controllers, parameters of the evolutionary process, such as the probability of including, deleting or changing synapses and neurons, can be modified online. Other such parameters are the average size of a population, the steepness of a selection function, and costs of neurons and synapses, etc.

For convenience, neurons of the controllers will be of the additive graded type with zero bias terms. The dynamics of the controllers is then given by

$$a_i(t+1) = \sum_{j=1}^{j} w_{ij} \cdot f(a_j(t)) + I_i, \qquad (1)$$

The transfer function is given by the hyperbolic tangent f(x) = tanh(x).

For simplicity, the controlled robot will be only able to move in one dimension, backwards or forward. Experiments with the resulting controller will then be tested on a physical robot. The physical robot platform we use for this approach is the GMD Musashi RoboCup goalkeeper [4]. This robot is a four-wheel robot, with a two wheel differential drive. It uses an omni-vision camera system, which is the only physical sensor that is used for our experiments. Data from this vision system are derived from blob detection [5] and include angle to the goal, angle to the ball, distance of the ball, etc.

The used simulator, DDSim [3], provides a 2 ½ dimensional simulation of wheeled robots. The interface and the data delivered by the simulated robot correspond to the physical robot, so that the effort to adapt the controller to the physical robot is reduced to a minimum.

The controller inputs are defined as follows; α_g is the relative angle of the goalkeeper to the goal, mapping [-180°:180°] onto [-1:1], l_g the relative width of the goal blob detected by the camera to the maximum possible width, mapped to [-1:1] where negative values correspond to a position close to the goal, α_b the relative angle of the goalkeeper to the ball, mapping [-180°:180°] onto [-1:1] , d_b the distance of the goalkeeper from the ball, mapping [0:700]cm onto [-1:1], where -1 denotes 700cm or not visible, v the velocity of the robot, mapping [-160:160]cm/s onto [-1:1]. The controller output m is the motor command, mapping [-1:1] onto [-160:160]cm/s.

3. Experimental Setup

The overall setting for the RoboCup Mid-Size League robots is a well defined play field [1]. To evolve goalkeeper behaviour we have to design an appropriate fitness function. To achieve this, we first define three different regions for the goalkeeper on the play field: the food place, the habitat, and the lethal area. For an omni-directional vision system, as used in the GMD RoboCup robot, the centre of the goal, extended by some radius r_g is the optimal waiting position (food place), as the maximal distance to the goal posts is minimised, and an approaching ball is visible from any direction and position of the robot. The goalkeeper is in the food place, if its coordinates (x_g, y_g) satisfy

$$(x_g - x_f)^2 + (y_g - y_f)^2 < r_f^2\ , \quad 0 < x_g \qquad (2)$$

where (x_f , y_f) is the centre of the food place, which also is the centre of the goal line. The habitat is the extended area, in which the robot is allowed to move. The area is chosen as an equivalent to the penalty area defined by the RoboCup rules [1]. Defending a ball outside of this area should be the task for a defender, not the goalkeeper. The functions defining this area are then given by

$$x_g \in [x_0, x_1]\ , \quad y_g \in [y_0, y_1] \qquad (3)$$

where {(x_0, y_0), . . . , (x_1, y_1)} denote the four corners of the penalty area. The lethal area is then defined by negation of the habitat.

The fitness is measured in terms of the energy E of the robot. Initially the robot has an energy depot E_0 from which it can live of. With E_t we will denote the energy level at the time t. Time is measured in discrete steps starting with 0, t = 0, 1, 2, . . . , (t_{max} - 1), where t_{max} is the maximal evaluation time of the robot in the environment. The energy depot is limited by a defined maximum of energy E_{max} the robot can store. With $\Delta e = E_{t+1} - E_t$ we will denote the energy difference from one time step to the next. The evaluation of the robot ends, if the maximal lifetime is reached, or if the energy level satisfies $E_t = 0$. If the robot moves, it consumes energy according to an additionally movement term $e_m > 0$, which will be subtracted from E_t. Thus the three areas defined above are characterised by the difference Δe in the following way: $\Delta e > 0$ if the robot is in the food place, $\Delta e < 0$, if it is in the habitat, and $\Delta e = -E_{max}$ if the robot is in the lethal area. In addition, the environment is also determined by the behaviour of the dynamical object, the ball. If the ball passes the goal line, the evaluation of the robot is terminated ($E_t = 0$). As the setup of the environment should be the same for every robot within one generation, a list of $B = \{b_0, . . . , b_{n-1}\}$ with $|B| = n$ different ball settings will be produced. A ball setting $b_i = \{x_i, y_i, \alpha_i\}$ is a defined starting point (x_i, y_i) with a starting angle α_i. The ball speed is set constant, but is high with respect to the robot speed for all balls. As discussed later, the result shows that this does not mean any restriction for the behaviour control of the robot. If the ball has passed the goal, or was reflected by the goalkeeper (non-kill situations) a new ball must be picked from the list of balls. The list of balls B is implemented as ring. The agent must be able to regenerate energy, after reflecting a ball, therefore no ball is presented to the goalkeeper for p_b cycles, after a ball is reflected.

To avoid the bootstrap problem [8] we use incremental evolution in the following sense. The first tasks presented are chosen to be solvable without much controller structure. The difficulty of the tasks increases, if more than one controller has an almost optimal fitness (see below) over more than one generation. The first generations of controllers face a rather simple task. It can be described as “find the food!”. Every robot in all generations is started at the corners of the goal. The maximal lifetime t_{max} of an agent is chosen so that simply passing the food place, i.e. moving straight forward or backwards, can increase the fitness. Staying in the food place will result in a high fitness. As soon as agents are evolved which solve this task, balls are shot in such a way that the evaluation of those robots is terminated, which only oscillate over width of the goal or simply stay in the food place. After this task is solved sufficiently, new balls are added to the list of balls, and the maximal lifetime t_{max} is increased, such that the difficulty of the task is increased.

A reasonable fitness function for the evaluation should be able to distinguish between agents, that did reach the maximal life time and agents that did not, as well as between robots, that were energy conserving and such that were not. Only using the number of time steps does not distinguish between robots, which reached the maximum evaluation time.

Some might have learned to be very energy conserving, so they deserve a higher fitness. Only using the remaining energy does not distinguish between robots that were terminated during the evaluation process, as they all have E = 0. But some might have lived longer than others (reflected more balls), and therefore deserve a higher fitness. To have the desired properties of distinction between individuals, we use both terms, energy and life time. The fitness function is then given by

$$F = \frac{t}{t_{max}} + \frac{E_t}{E_{max}}, \qquad F \in [0, 2].$$

An agent, which reached t_{max} will have the energy term added to its fitness, compared to an agent that reached t_{max} - 1. Consequently it will be able to produce significantly more offsprings. This results in a high selection pressure to reach the maximal life time. The controller has 5 input neurons and 1 output neuron (see sec. 2). The evolutionary process starts with only the input and output neurons. No connections or hidden neurons are set. The energy level was not chosen to be an input to the neural network, as the physical robot has no energy depot equivalent to the setup describes above. The inputs and outputs are chosen according to the input and output signals of the physical and simulated robot [3].

4. Results & Analysis

Among the results of the evolutionary process was a small neuro controller that did not use all of the presented inputs. In fact, it only uses the relative ball angle α_b and the relative goal angle α_g (see fig 2, left) to achieve maximal fitness. The robot generated by this controller, shows a behaviour that is optimal concerning the setup and the goals defined before. The structure evolution presented a varity of different neural network tolopologies, with a higher number of hidden neurons and synapses. The neural network presented here outperformed or showed equal performance compared to the other networks, but with a minimal structure. Therefore this networks was choosen for presentation. The properties of this controller can be described as follows: *Goalkeeper*: The controller was able to defend all balls that headed towards the goal (those from the ball list, as well as those manually positioned and moved). *Penalty Area:* The controlled goalkeeper never leaves the penalty area, even if balls pass the goal in a large distance. *Positional Play*: The controlled goalkeeper did not just follow the ball, but showed a good positional play. The goalkeeper positions itself almost at the line, connecting the centre of the goal with the centre of the ball (see fig 3). In the following we will discuss the formal analysis of the controller, followed by experiments that represent the most common situations on the play field. For the formal analysis we will discuss the dynamics of the controller, which is given by

$$a(t+1) = w_g \alpha_g + w_b \alpha_b + w_s \tanh(a(t)) \qquad (5)$$

where w_b is the weight from the ball angle input neuron, w_g is the weight from the goal angle input neuron, and w_s is the self-connection of the output neuron. Having a sub-critical self-connection ($w_s < 1$) and the hyperbolic tangent as transfer function, the isolated output neuron always has the origin as a stable fix point a* = 0 [2]. This fixed point a* = 0 corresponds to a resting goalkeeper. To derive the values of α_g, α_b for which the goalkeeper will stop, we solve the fix point equation and finally get

$$\alpha_g = -\tfrac{w_b}{w_g}\alpha_b \ . \qquad (6)$$

The solutions of equation (6) describe configurations for which the robot stops. Figure 1, right, shows the relationship between the angle of the ball α_b and the distance y_g of the goalkeeper from the centre of the goal, which is given by the equation

$$y_g = x_g \cdot \tan(-\frac{w_g}{w_b} \cdot \alpha_g)$$

with x_g = 85 [cm] denoting the fixed distance of the goalkeeper from the goal line. It is seen, that the goalkeeper can not leave the penalty area for any valid value of the relative ball angle α_b. In the following we will compare this formal analysis with experiments that represent extreme and common situations on the play field. The plot in figure 2 shows the inputs and the output over 1000 time steps with five different intervals where the ball was placed by hand. The first section of the recorded data corresponds to the reaction of the goalkeeper, if the ball is placed in the upper corner of the field. The ball was first not visible and then suddenly appeared in the vision field of the robot. This can happen during a real game, as the other robots often gather around the ball, and therefore cover it completely. The strong deflection is the result of the recurrent connection. The following oscillations result from oscillations of the input data, in this case the goal angle α_g. This oscillation was identified as an artefact of an older version of the simulator, which was used here. One can see, that the goalkeeper will not leave the goal after reaching some maximum goal angle α_g. The same holds for the other side (ball position 2), where the goalkeeper will also not leave the penalty area. The larger amplitude of the oscillations are due to the larger amplitude of the oscillating input values delivered by the simulator. In section 3 there are no oscillations in the input data, and it can be seen, that the robot then takes and holds it's optimal position. The high peek of the controller output is again the result of a non visible ball, that suddenly appears in the vision field of the robot. The same situation, but mirrored can be observed in section 4. In the last section the ball is placed in front of the goalkeeper, and causes almost no reaction of the robot.

5. Discussion

An evolutionary robotics approach was used to develop a neural control for an effective behaviour of a RoboCup goalkeeper. Motivated by Artificial Life arguments, the reference to an energy reservoir of the agent led to the separation of the environment into nutritious and hostile areas. This allowed the definition of a simple fitness function, which nonetheless coded the desired behaviour in a very effective way. Although the problem was reduced to that of driving a goalkeeper with only one degree of freedom, the result of the applied evolutionary techniques is remarkable in two ways. First, the evolved neurocontroller is structurally of astonishing simplicity: it uses no internal neurons and only a self-connection of the output neuron. This made an analysis of the underlying functionality very simple. It turned out, that the controller implements the strategy, that the goalkeeper roughly has to stay on the line from the ball to the centre of the goal. Second, evolution showed that for an effective goalkeeper behaviour not all of the available sensor information was necessary: Only two of the supplied five inputs where used, namely the relative angles to the ball and to the goal. Nonetheless, the controller is able to handle balls with different speeds starting at all allowed field positions. Especially, for this control no prediction of the ball position or speed is needed.

Experiments with the physical robot of the GMD-Musashi RoboCup-team showed similarities as well as differences to the simulations. As expected, the robot did not hold its 1-dimensional orientation; because of the slip of the wheels it turned towards or away from the goal. Despite this fact, the physical robot showed a behaviour comparable to that observed in simulations. But the physical robot also showed oscillations, which turned out to be the result of data bufferring and delays resulting from the robot's inertia.

In forthcoming experiments the simulation will include also the physical properties of the robot. Furthermore, the new goalkeeper is assumed to move in two dimensions, realising an optimal positional play.

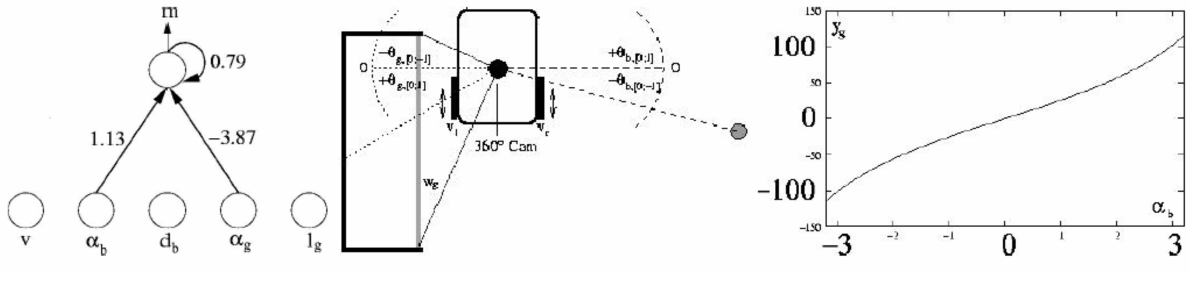

Figure 1: Left: The controller has 5 inputs, and 1 output (see sec. 2). Centre: The two inputs used by the controller. Right: The relationship between the angle of ball α_b the distance y_g. The

angle α_b is given in rad, the distance y_g in cm. The goal is 200cm wide, the penalty area is 300cm wide.

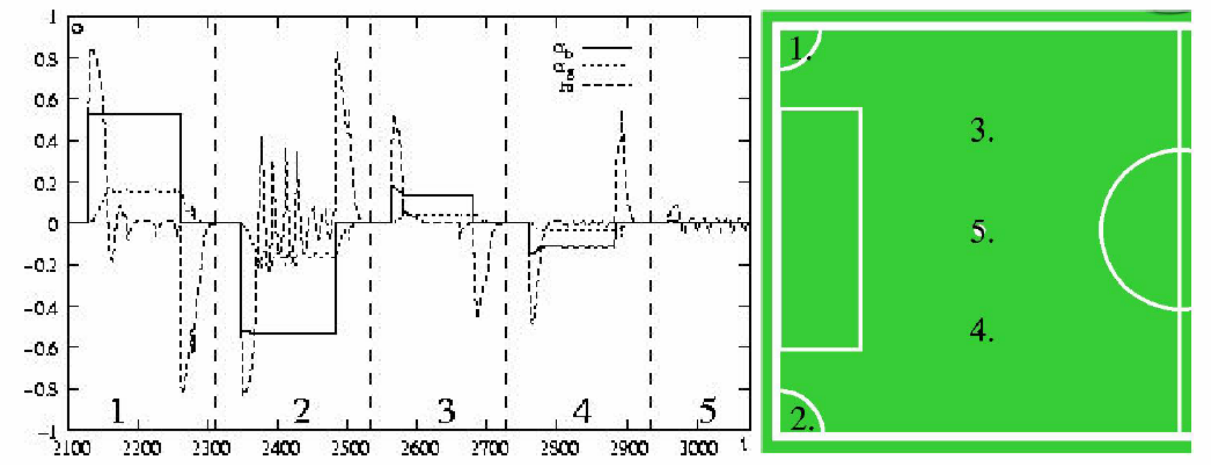

Figure 2 The left plot shows the inputs α_g, α_b and the output m for different situations over 1000 time steps. The numbers correspond to the positions on the field on the upper right figure.

References

[1] Robocup official site. http://www.robocup.org, 2003.
[2] Pasemann, F.,Dynamics of a single model neuron, *International Journal of Bifurcation and Chaos*, **2**, 271-278, 1993.
[3] A. Bredenfeld, T. Christaller, W. Göhring, H. Günther, H. Jäger, H.-U. Kobialka, P.-G. Plöger, P. Schoell, A. Siegberg, A. Streit, C. Verbeek, and J. Wilberg. Behavior engineering with 'dual dynamics' models and design tools. Lecture notes in computer science, pages 231 – 242. RoboCup-99: Robot Soccer World Cup III, 2000.
[4] A. Bredenfeld, T. Christaller, J. Hermes, G. Indiveri, H. Jäger, H.-U. Kobialka, P.-G. Plöger, P. Schoell, and A. Siegberg. GMD-Robots. Lecture notes in computer science, pages 579 – 582. RoboCup 2000, 2001.
[5] J. Bruce, T. Balch, and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. volume 3, pages 2061–2066. In Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00), October 2000.
[6] D. Floreano and J. Urzelai. Evolutionary robotics: The next generation. In T. Gomi, editor, Evolutionary Robotics III. AAI Books, Ontario (Canada), 2000.
[7] M. Hülse and F. Pasemann. Dynamical neural Schmitt trigger for robot control,. In J. Dorronsoro, editor, Artificial Neural Networks - ICANN 2002, volume LNCS 2415, pages 783–788, Berlin, August 2002. Springer Verlag.
[8] S. Nolfi and D. Floreano. Evolutionary Robotics. MIT Press, 2000.
[9] Hülse, M., Lara, B., Pasemann, F., Steinmetz, U., Evolving neural behavior control for autonomous robots, in: G. Dorffner, H. Bischof, and K. Hornik (Eds.): ICANN 2001, LNCS 2130, pp. 957--962, 2001
[10] F. Pasemann, U. Steinmetz, M. Hülse, and B. Lara. Robot control and the evolution of modular neurodynamics. Theory in Biosciences, 120:311–326, 2001.