A Very Short Story About Autonomous Robots

Gerald Steinbauer[*], Roland Koholka[#], Wolfgang Maass[*]
[*] Institut für Grundlagen der Informationsverarbeitung, Technische Universität Graz
[#] KNAPP Logistik Automation GmbH, Graz

Abstract

Machine learning for autonomous mobile robots is a very interesting but also very difficult problem. In this domain you need robust, fast and efficient learning algorithms. The combination of machine learning and mobile robots allows us to create machines whose performance surpasses both explicitly programmed robots and humans. Although humans can learn, their sensors and actuators are in several aspects inferior to those of robots. On the other hand a non-learning robot can only perform tasks where all the individual difficulties and complexities can be anticipated by the programmer of the robot.

In this short article we discuss both a completed project, and a new one that has just started. The first project produced a robot that applied learning to improve his minigolf skills. The new project is the creation of the first Austrian team of autonomous robot soccer players that will compete in the robot soccer world championship RoboCup[1].

Machine learning for real robots

If you work on machine learning for real robots you may ask yourself why machine learning for real robots is so much more difficult than for other applications, such as simulated robots, optimization or classification. There are three main reasons for that. First, everything in the real world is nondeterministic and noisy. You could not rely absolutely on what the sensors of the robot perceive or on what the actuators of the robot may do. For that reason the learning algorithms have to be robust against this insufficiency of the robot. The second reason is that most machine learning methods require hundreds or thousands of training examples. But a real robot cannot have the luxury of training for weeks or months. The third reason is the limited computing power of a mobile robot. It is not very satisfying if a robot stops for ten minutes in order to think about its next decision. Hence, extremely efficient learning algorithms are needed. All these problems have their own charm, but they can also be very frustrating in the end. In spite of all this, the work with machine learning and real robots is quite rewarding. To see a robot show a surprising accurate behavior for a certain task, where explicit programming of such behavior was nearly impossible, is very satisfying.

From Robot Minigolf ...

In the earlier project with real robots we worked on a task for an autonomous robot that requires learning insofar as a solution of this task by a non-learning robot is inconceivable (see [1]). On the other hand this task is also too difficult to be solved by a human. People can learn, but they do not have the mechanical skills which this task requires. The task had been posed in the form of a student competition at the Technische Universitaet Graz. It can be outlined as follows. A 2x5 m white platform – surrounded by a black wall – had been divided by a black line into a release zone of about 1 m length and a target zone of about 4 m length (see Figure 1). For each instance of the task one out of a large variety of green colored hills was placed at an arbitrary position – but at least 40 cm away from all walls – into the target zone. The hills were formed out of different kinds of hardening or non-hardening resins. These hills had a relatively rough surface and all kinds of odd shapes (diameters 30-60 cm), but they all had a round dip on the top of about 2 cm depth, with diameters ranging from 8 to 12 cm. The task was to accelerate and release a red billiard ball (diameter 5 cm, weight 142 g) in the release zone on the left part of the platform so that it comes to rest in the dip on top of the hill. To solve this task, the ball had to be released with just the right speed v and angle α. For most hill positions the set of parameters $\langle v, \alpha \rangle$ that solved the task was so small that even an experienced human typically needed 40 or more trials before the first successful shot. The number of trials needed for a second successful shot was not significantly smaller, indicating that in spite of all experience a human can solve this task essentially just by chance.

After each unsuccessful trial the robots in the competition had to find the ball, move it back to the release zone, and initiate the next shot. All this – just as the trials themselves – had to be done completely autonomously, using only on-board sensors, computing devices and power

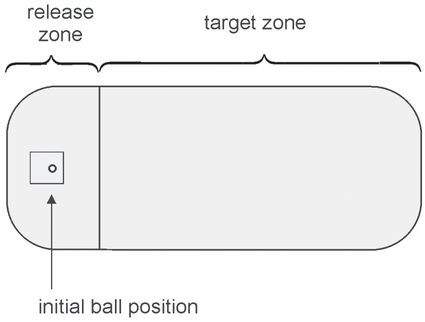

Fig.1: *The minigolf playfield.*

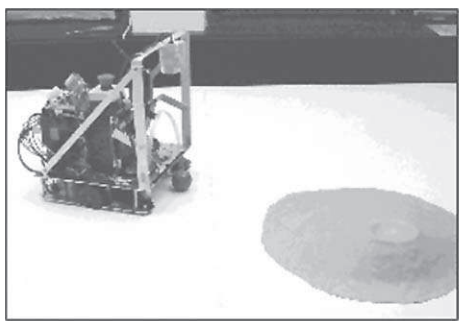

Fig. 2: *Robot Oskar while recapturing the ball.*

1 This project is partially supported by Knapp Automation and Logistics, Siemens AG, and JUMPtec.

supply. There were no interventions from the outside allowed. The performance measure for the competition was the total time needed by a robot until he had succeeded three times for the current hill and hill-position, averaged over a fair number of different hills and hill-positions.
The winner of the competition was the robot Oskar (see Figure 2) because of its simple and reliable hardware/software design and its fast and robust learning algorithm. Oskar moves via a differential drive powered by stepping motors. This enables Oskar to reposition himself very precisely in the release zone using visual markers (±1mm displacement, 0.2° angle error). This fact is important for a meaningful interpretation of successive failed trials, as well as for producing repetitions of successful trials. Oskar uses two cameras as its main sensors. One camera looks ahead (used mainly for ball tracking), the other camera looks orthogonally down in front of the robot (used mainly for obstacle avoidance). Oskar accelerates the ball by hitting it with a hammer that can be pulled up to variable heights, thereby allowing control over the initial speed *v* of the ball. Oskar catches the ball by lowering a flexible „lasso“. He transports the ball back into the bay in front of the hammer with a cute body movement initiated by his two wheels („hip swing“). For further details about Oskar see [2], [1], and [W1].

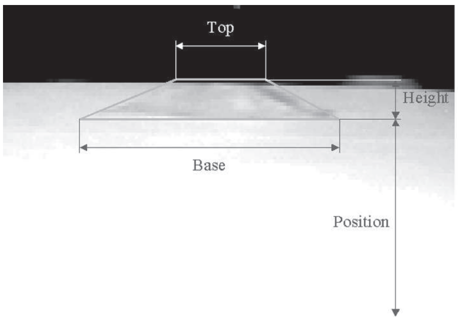

Fig.3: *A typical hill, as seen by the onboard camera of Oskar. The trapezoid approximation provides the inputs for the neural network.*

The learning of Oskar is implemented as follows. Oskar uses both a longterm- and a shortterm learning algorithm. In order to compute the initial speed *v* for a new instance of the task he uses longterm learning via a sigmoidal neural network (MLP) with 4 hidden units. This neural network absorbs memory from positive experiences in the past. The inputs to the neural network are the coordinates of the 4 corner points of a trapezoid (see Figure 3) that approximates the segmented video image of the hill, recorded from the starting position. The neural network is trained via backprop, with training examples provided by preceding successful trials. The neural network started to make useful predictions for an appropriate initial speed *v* after it was trained with data from 60 successful trials.
The other parameter a for the first trial on a new hill is simply computed by aiming at the center of the upper horizontal line segment („Top“) in the trapezoid that approximates the hill. Since Oskar's first trial is likely to succeed only for very simple hills, the key point of his operation is his capability to learn via shortterm learning autonomously from preceding unsuccessful trials for the same instance of the task. The shortterm learning is inspired by the two principles of „trial and error“ and the classical binary search. From the mathematical point of view it is a two-dimensional searching problem.
First Oskar draws the trajectory of the ball for each shot (the trajectory is extracted from camera images recorded during the shot). Then he classifies the trajectory by a hand-optimized, heuristic classification algorithm into one of the following classes:

1. ball went too much to the right
2. ball went too much to the left
3. ball reached the dip, but went through it
4. ball rolled back from the hill
5. success
6. anything not fitting into classes 1-5

This classification is used as a feedback for the binary search, although it can be erroneous due to insufficiencies of the image processing and the classification algorithm. Therefore we had to adapt the classical binary search to be robust against erroneous feedback. This robustness is achieved by extra mechanisms. In contrast to the ordinary binary search, we halve the search interval only if we see „opposite“ classification pairs (class pair 1/2 or class pair 3/4) in successive trials. Furthermore, in some cases the search interval may be doubled again, enabling a recovery after erroneous answers. Fortunately each of the two classification pairs is relevant for just one of the two binary searches. The classification pair 1/2 triggers the binary search for the angle a. The classification pair 3/4 triggers the binary search for the velocity *v*. In case of classification 5 and 6 neither the angle a, nor the velocity *v* are changed. In the latter case we are not able to extract a relevant feedback. In the former case we have already found a solution. Furthermore, if this solution is stable enough, it provides a new training example for long term learning. This mechanism collects positive experiences for the robot. An exact description of the learning algorithm can be found in [1] and [2].

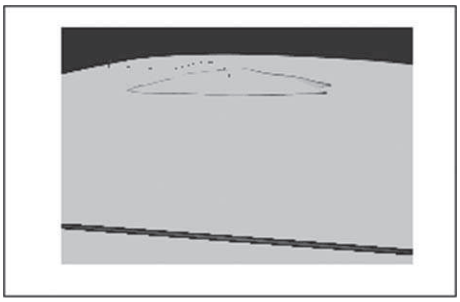

Fig.4: *Typical trajectory, classified to class 2.*

This simple learning algorithm works surprisingly fast and robust for the given task. Oskar needs about 1 to 8 trials for various hills and positions, while humans need about 40 trials for the same task. Oskar's hardware is also very robust because of its simple design. He had been running continuously 9 hours a day during 6 months of a major exhibition, the Steiermaerkische Landesausstellung in Graz. Currently Oskar is part of the permanent collection of the famous Ars Electronica Center in Linz [W2] and he is still working every day (Sundays included).

... to Robot Soccer

After our robot Oskar had learned to play minigolf, we got a little bored. So we got interested in a new sport for our robot, namely robot soccer. We decided to design and build a team of new autonomous mobile robots, that are able to compete in the RoboCup Middle-Size League World Championship 2003 in Padua, Italy. Since a wide range of knowledge is needed to build a team of autonomous mobile robots, many institutes of the Technische Universitaet Graz are involved in this project. A complete list can be found on the project's homepage [W3].

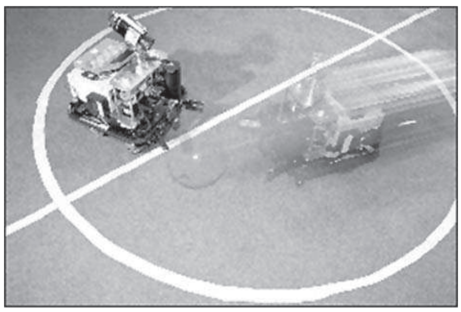

Fig.5: *Players of the GMD Middle Size League Team.*

The Robot World Cup Initiative [W4] is an attempt to foster artificial intelligence and intelligent robotics research by providing a standardized problem that poses a tough challenge for several scientific disciplines and technologies. The first RoboCup competition [3] was held 1997 at the IJCAI in Nagoya. The interest in RoboCup and the number of participating teams increases every year. Until a robot team is actually able to perform a soccer game, various technologies have to be incorporated, including multi-agent cooperation, strategy acquisition, real-time reasoning, machine learning, robotics, and sensor-fusion. Contrary to other robots, which are optimized for a single heavy-duty task, robot soccer is a task for a team of cooperative fast-moving robots in a dynamically changing environment. The RoboCup is organized into several leagues. They differ in several aspects: simulated or real robots, the types of sensors (global or individual), and the size of the robots. So everybody can find the optimal platform for their research.

The games in the simulation league run on the RoboCup soccer server [W5] which is accessible by the public. The RoboCup soccer server provides a standard platform for research on multi-agent systems. The soccer server simulates the player, the ball and the field for a 2D soccer match. Up to 22 clients (11 for each team) connect to the server, each client controlling a single player. The client sends low level commands (dash, turn or kick) to be executed (imperfectly) by the simulated player he is controlling. The soccer server simulates the (imperfect) sensing of the player, sending an abstracted interpretation to the clients. This league has already reached a high level in intelligent behavior, learning, and team play [4].

Small-size robot teams consist of up to 5 robots, with each robot fitting into an area of 180 cm^2.

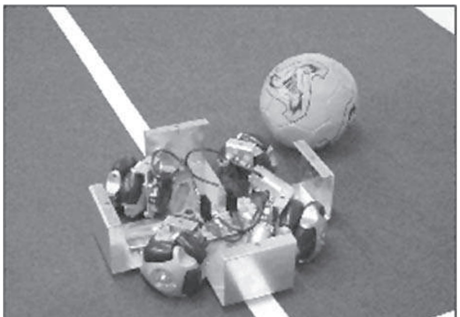

***Fig.6:** The prototype of our driving unit.*

The robots play on a green-carpeted table-tennis-sized field with sloping walls. The rules permit a camera to be fixed above the field and connected with an off-field computer for a global vision system. This system is used to track the players, the opponents and the ball. During the game the robots use wireless communication to receive tracking information from the off-field computer, as well as commands or strategic information. No human intervention is allowed except for interpretation of the referee's whistle. The teams in this league have developed interesting strategies for cooperative behavior, but they are still not at the level of the simulation league.

In the middle-size RoboCup league, teams of four roughly 50x50x80cm sized robots compete. The field size is currently 9m x 5m. The major difference to the small-size league, in addition to the size of the robots and the field, is that no global vision of the field is allowed. Thus the robots have to rely totally on their own sensors, including vision. The robots are fully autonomous, i.e., their sensors, actuators, power supply and computational power are on-board, and no external intervention by humans is allowed, except to insert in or remove robots from the field. External computational power is allowed, although most teams do not use it. Wireless communication between the robots and/or with the external computer is also allowed. As in most of the other leagues, relevant objects are distinguishable by their color: the ball is orange, the goals are yellow and blue, the robots are black, the walls are white, the robot markings (to distinguish the teams) are magenta and light blue. The middle-size league provides a serious challenge for research disciplines such as multi-robot cooperative teams, autonomous navigation, sensor fusion, vision-based perception, and mechanical design, to name only a few.

In the last year the community has decided to remove the walls around the field for the middle-size league, as another step towards a more realistic soccer game. Since now the ball and also the robots are able to leave the field, the demands on the robots regarding ball handling, perception, and strategy increase. This may be a realistic chance for a new upcoming team to participate successfully, because also the established teams have to change their paradigms.

Currently artificial intelligence and machine learning do not yet play in this league such an important role as, e.g., in the simulation league. The problems caused by perception (mainly vision), self-localization, mechanical and electronic design (ball handling and robot drives) are still dominating, and make it more difficult to implement adaptive, intelligent, and cooperative behavior.

In view of this fact we have divided our RoboCup project into two sections, which hopefully will meet each other again in the end. One section is working on a new powerful robot platform, which should provide optimal perception and optimal dynamics for playing soccer, because commercial robot research platforms are still not optimal for playing soccer. By studying games that were played during official tournaments in the last year, we saw that a platform which is fast and flexible in movement is very important for a successful dynamic soccer play, and provides a chance to outperform the opponents. So the main design goal for our platform is the development of a fast omni-directional driving unit. In the 4th Robotic Soccer World Championship in Melbourne in 2000 the Golem Team [5] from the University of Padua showed an omni-directional drive for their robots with three universal wheels, each driven by a separate DC Motor, mounted at the vertexes of an equilateral triangle. This arrangement allows the robot to control all three degrees of freedom in the plane (Δx, Δy, $\Delta\varphi$) simultaneously. In order to increase the stability of the robot platform, especially during a rough soccer play, we adapt this configuration to a model with four wheels. Our configuration of the four motors and wheels is shown in Figure 6. The other section is working on a simulator for the upper platform, an adaptation of the Webots simulator [W6], which should give us the opportunity to implement and investigate high-level behaviors and learning approaches even before the real platform has reached a final stable state (only the kinematics and sensor inputs of the robot must be known in advance). This gives us the possibility to simulate and study higher level behavior and machine learning algorithms on our platform in a sufficient realistic manner while we are still working on the implementation of the real platform. We hope that we can transfer higher level strategies and learned behaviors back to the real robot platform.

Since we are now at a very early stage in the project, our currently used agent architecture is quite simple but easy to handle and to understand (see Figure 7). It consists of a perception module, a behavior selector and a set of basic behaviors or skills. The perception module processes all sensory inputs (mainly vision and odometry) and provides a model

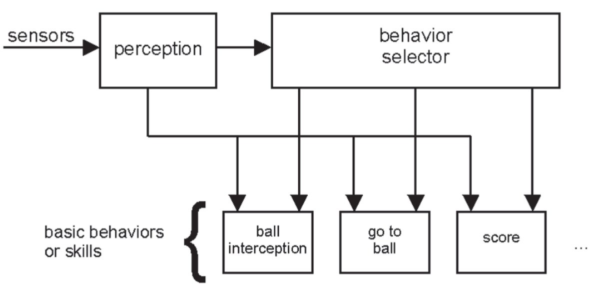

***Fig.7:** Agent architecture of our soccer robots.*

of the state of the robot and its environment (mainly the position of all the robots and the ball). Based on this state information the behavior selector selects one behavior from a given set of behaviors or skills which the robot is able to perform. The selected behavior is executed for a given short period of time. The set of the basic behaviors consists of very fundamental skills the robot can perform.

This architecture has three main advantages. First, it is simple to understand and to implement, which is very important in view of the early state of the project in order to get a feeling for the potential of our platform. Second, the robots have only to provide a small set of „primitive“ behaviors or skills. By having the behavior selector combine these basic behaviors a relatively complex overall behavior could be achieved. This approach builds on the subsumption architecture by R. Brooks [6]. The behavior selector is currently designed and optimized by hand. The third advantage is that the simple basic behaviors or skills can also be learned and autonomously optimized by the robot, using for example reinforcement learning. This approach is discussed in the book of P. Stone [4] for the simulation league.

Conclusion

We have tried to show that the work with autonomous mobile robots and machine learning for such robots is difficult, sometimes frustrating, but most of the time fascinating and satisfying. With the robot Oskar we had demonstrated that the combination of machine learning and a simple robust robot platform may be able to fulfill tasks where humans fail. This was a very surprising experience to us. Finally, we hope that we will be able to transfer the „spirit“ and the success of Oskar to our new robot soccer team.

References

[1] W. Maass, G. Steinbauer, and R. Koholka. Autonomous fast learning in a mobile robot, to appear 2002 in: *Lecture Notes in Computer Science*, Springer-Verlag.

[2] Gerald Steinbauer. *Machine Learning for an Autonomous Mobile Robot*. Diploma thesis. Technische Universitaet Graz, Institute for Theoretical Computer Science, 2001.

[3] H. Kitano, editor. *RoboCup-97: Robot Soccer World Cup I*. Number 1395 in Lecture Notes in Artificial Intelligence. New York: Springer, 1998.

[4] Peter Stone. *Layered Learning in Multiagent System: A Wining Approach to Robotic Soccer*. The MIT Press, 2000.

[5] Paolo de Pascalis et al. *Golem team in middle-sized robots league*. In RoboCup 2000: Robot Soccer World Cup IV. Number 2019 in Lecture Notes in Artificial Intelligence. Springer, 2001.

[6] R. A. Brooks. *A robust layered control system for a mobile robot*. IEEE Journal of Robotics and Automation, Vol. 2, No. 1, March 1986, pp. 14-23.

Links

[W1] http://www.igi.tugraz.at/maass/robotik/oskar: The Homepage of the robot Oskar.

[W2] http://www.aec.at: Ars Electronica Center, Linz, Austria.

[W3] http://www.robocup.tugraz.at: The RoboCup Project of the Technische Universitaet Graz.

[W4] http://www.robocup.org: The RoboCup Federation.

[W5] http://sserver.sourceforge.net/NEW/: The RoboCup Soccer Server.

[W6] http://www.cyberbotics.com/webots/: The Webots Robot Simulator. ■

Dipl. Ing. Gerald Steinbauer, computer scientist, studied Telematik at the Technische Universität Graz, together with Dipl. Ing. Roland Koholka winner of the first robot competition 1995 of the IEEE-Student Branch at the TU Graz, and of the robot competition *Robotik 2000* at the TU Graz, heads together with Dipl. Ing. Roland Koholka the development of a RoboCup team at the TU Graz.

Dipl. Ing. Roland Koholka, computer scientist, currently at Knapp Logistik GmbH, studied Telematik at the Technische Universität Graz, together with Dipl. Ing. Gerald Steinbauer winner of the first robot competition 1995 of the IEEE-Student Branch at the TU Graz, and of the robot competition *Robotik 2000* at the TU Graz, heads together with Dipl. Ing. Gerald Steinbauer the development of a RoboCup team at the TU Graz.

Prof. Dr. Wolfgang Maass, computer scientist and mathematician, Ph.D. under Prof. K. Schütte (Mathematical Logic) at the University of Munich, 1986-1991 Professor at the University of Illinois (Chicago), since 1991 Professor at the TU Graz, since 1992 Head of the Institut für Grundlagen der Informationsverarbeitung der TU Graz. Homepage: http://www.igi.tugraz.at/maass