Team WROCŁAW'2002

Paweł Rogaliński, Marek Piasecki

Institute of Engineering Cybernetics, Wrocław University of Technology,
ul. Janiszewskiego 11/17, 50-372 Wrocław, Poland
pawel@ict.pwr.wroc.pl, piasecki@ict.pwr.wroc.pl

1 Introduction

So far, the team representing Wrocław University of Technology is the only team representing Poland that has participated in official Robocup simulation league competitions (Robocup European Championship 2000, Robocup German Open 2001). Presently, team WROCŁAW'2002 comprises two lecturers (Marek Piasecki and Paweł Rogalinski) along with students of Electronics Department (Radosław Rudek, Paweł Trociński, Mariusz Tywoniuk, and others).

Team genesis: research on multi-level parametrical and structural evolution

Robocup simulation league is an interesting example of multiagent system which facilitates testing and comparing various techniques of creating algorithms controlling agents operating in an environment having features similar to the "natural world". Trials have been undertaken at Institute of Engineering Cybernetics, Wrocław University of Technology, to create such algorithms through evolution [1].

Initial research works utilising evolutionary techniques for creating Robocup player control algorithms, were undertaken in the year 1999. They constituted automatic optimisation of player control algorithm parameters using the conventional genetical algorithm [2]. A simple reactive algorithm performing low-level behaviour such as "run to the ball and shoot towards the goal", was then used as a test task. Program structure was fixed and set a priori by the programmer. Only the parameters were subjected to optimisation. This method enabled fine-tuning the program to current characteristics of the objects (player and ball) but required setting general structure of the algorithm. Basic inference of this stage was to indicate the need for automatic program structure generation.

In the second stage of testing, genetical programming algorithm was used for automation of reactive controller programming, enabling evolutionary creation of program structure represented in the form of instruction tree. Due to very large size of solution space and long duration of experiments, it became necessary to introduce rough digitization of parameter space. The effect of such an approach was to accelerate the process of evolution to a period of few days but at the cost of sub-optimality of solutions obtained. Obtaining an optimal solution requires alternative application of structural evolution (to generate the general form of algorithm) and parametrical evolution (to locally tune this general algorithm).

Further research works concerned the possibility of simultaneous evolution of both structures as well as parameters of the program [3]. We assumed that an important element is the capability of entering the initial knowledge (given by an expert) which could direct and considerably accelerate the course of evolution process. In our experiments, we attempted to introduce the expert's knowledge in three ways:

1. Evident setting of whole or part of initial population (e.g. by generating programs from sections of code acknowledged by the expert as the most interesting).
2. Utilise the directed recombination (e.g. selection of intersection point in such a way so as to avoid breaking up of useful sections, and focusing mutations on sections generating weaker evaluations).
3. Introducing program representation that would disable creating solutions incorrect from the expert's point of view (e.g. eliminating endless loops).

Conventional programming techniques used by human clearly stress the usefulness of structural or objective approach in cases of larger applications. On the basis of analogy, in genetical programming it seems necessary to introduce structuralization of evolution which enables separate generation of subprograms (procedures) and programs utilising such procedures [4]. Such an hierarchical evolution should be conducted simultaneously at all levels and would require program coding by means of sub-tree file representing the main program and constituent procedures.

Encouraged by the interesting results of research works conducted, we took up the challenge of competing in world championships RoboCup 2002. Unfortunately, the team software WROCŁAW'UT, utilised for scientific experiments (e.g. genetical programming of player behaviour) turned out in practice to be too "weak" in confrontation with game level of current RoboCup top teams. Therefore, the software of team Wrocław'2002 notified for the games is a compromise between the results of scientific works and the clever tricks created ad hoc, so as to meet the relevant rules and maintain minimum game level.

Team WROCŁAW'2002

Our team consists of ten homogeneous soccer players utilising the same general control algorithm and the eleventh specialised player performing the role of the goalkeeper. General structure of the program controlling our players is presented in diagram number 1.

The code of communication module is directly imported from the source of the team CMUnited'99. Materials provided by this team were also utilised while creating the first versions of Geometrical and Parametrical World Modelling module. Remaining player elements were created from scratch by our team.

The crucial point of our architecture is parallel connection of several low level reaction modules and high level deliberative action planning. Actions elaborated by various modules are subsequently evaluated and arbitrated by superior Task Manager.

At this stage of development, both approaches (reactive and deliberative) are applied only in action planning for single player. Cooperative sequences and deliberative planning for small formations are still under construction (marked with dashed line).

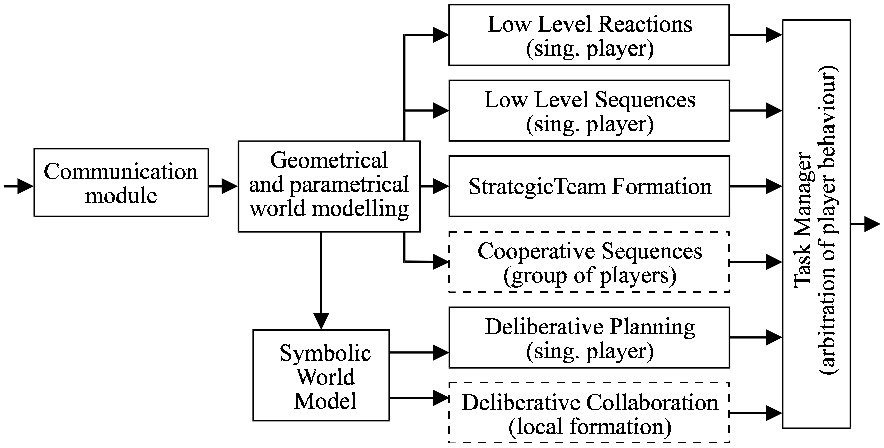

Fig. 1. Basic control modules of our soccer player

Included in Low Level Reactions module are all standard player behaviours (e.g. return to own half after goal) ensuring compliance of player's game with rules of Simulation RoboCup League. Also provided in the same module are specific hand-coded tricky reactions implementing low-level skills of the player. In the first approach these skills have been created by evolution. Unfortunately, the quality of reactions obtained in this way were unsatisfactory (in comparison with the games of the best teams). Ultimately, to equalise the chances of our team, we decided to implement them in the form of heuristic rules (decision tree) controlling single-step action of the player.

Low Level Sequences module implements multistage actions of individual player enabling precise or strong kicking of ball as well as ball dribbling to given target point. Sequences in this module are implemented in the form of Finite State Machines.

Deliberative Planning module utilises an additional symbolic model of the world representing the immediate surroundings of the player in the form of rosettes of free sectors (corridors between players of the opposite team). Free spaces inside these sectors are approximated by means of simple geometrical shapes. Next, each of these areas is valued with respect to: its size, distance from opponent's goal, number of teammates inside the area, etc. Finally, the set of central points of these areas generates the set of potential actions consisting in: dribbling the ball along a selected corridor or emergency kicking away the ball into the middle of a "safe" area. This same technique also enables creating simple group actions like "planning the place of expected pass" through modelling and analysis of free sectors around other teammates.

Strategic formation module generates default formation (arrangement of positions of players on the field) depending upon the situation on the field. At that moment, we utilise two basic formations (attack, defence) selected on the basis of which team is in possession of the ball. Default position of the player is parameterised by the current formation, position of the ball on the field and position of the player with respect to the ball.

Weaknesses

The greatest weaknesses of our team are: primitive technique of modelling the surroundings and lack of planning the multistage game of a formation of players.

The surroundings model utilised by us is to a large extent borrowed from the team CMUnited'99, which enabled us to considerably accelerate preparation of code of the first players. Unfortunately, this model was prepared mainly from the angle of local reactive planning realised by means of neuron-network type approximators. Utilisation of this same simple model for reactive and deliberative planning causes that our player is losing information too rapidly about the whole surrounding (same modelling process of short and long term parameters). In other cases model changes do not keep pace with changes in the immediate surroundings.

We therefore undertook the decision to split the modelling process into short and long memory. Near completion are works on advanced model extracting long-term characteristics of rival's game along with heuristic forecasting of behaviour of teammates (who are beyond the visibility sector of the player).

We have not yet been able to program long-term control of local formation of several players which would enable planning multistage actions comprising a sequence of passes ending with a shot to the goal. This aspect is the subject of very intensive works.

Conclusion

In this description we have presented the genesis and actual state of development of our team WROCŁAW'2002 with which we intend to participate in Simulation League competition RoboCup 2002. The basic code of this team was created to perform experiments with evolution and genetical programming of control algorithms. The final version is a combination of part of results of these experiments as well as manually coded heuristic behaviours created in conjunction with junior students of the Electronics Department.

References

1. Piasecki, M., Rogaliński, P.: Testing New Computational Paradigms in the Environment of RoboCup. (*in polish*) Proc. of I National Conference KOWBAN'99, Polanica Zdrój (1999)
2. Piasecki, M., Rogaliński, P.: Evolutionary Creation of Control Algorithms for RoboCup Players. (*in polish*) Proc. of XVIII National Conf. on Polioptim. & CAD, Mielno (1999)
3. Piasecki, M., Rogaliński, P.: Genteic Programming and Tuning of Control Algorithms for Processes with Incomplete Information. (*in polish*) Proc. of III National Conf. MSK (2001)
4. Piasecki, M., Rogaliński, P.: Computer Analysis of Robot's Behaviour in RoboCup Competition. (*in polish*) Proc. of 18 National Conf. on Polioptim. & CAD, Mielno (2000)