Strategy Description for Mobile Embedded Control Systems Exploiting the Multi-agent Technology

Vilém Srovnal¹, Bohumil Horák¹, Václav Snášel², Jan Martinovič²,
Pavel Krömer², and Jan Platoš²

¹ Department of measurement and control, FEECS, VŠB-Technical University of Ostrava,
17.listopadu 15,CZ-708 33 Ostrava-Poruba, Czech Republic
{vilem.srovnal, bohumil.horak}@vsb.cz

² Department of computer science, FEECS, VŠB-Technical University of Ostrava, 17.listopadu
15,CZ-708 33 Ostrava-Poruba, Czech Republic
{vaclav.snasel, jan.martinovic, pavel.kromer.fei,
jan.platos.fei}@vsb.cz

Abstract. Mobile embedded systems are a part of standard applications of distributed system control in real time. An example of a mobile control system is the robotic system. The software part of a distributed control system is realized by decision making and executive agents. The algorithm of agents cooperation was proposed with the control agent on a higher level. The algorithms for agents realized in robots are the same. Real-time dynamic simple strategy description and strategy learning possibility based on game observation is important for discovering opponent's strategies and searching for tactical group movements, simulation and synthesis of suitable counter-strategies. For the improvement of game strategy, we are developing an abstract description of the game and propose ways to use this description (e.g. for learning rules and adapting team strategies to every single opponent).

1 Introduction

A typical example of a distributed control system with embedded subsystems is the controlling of physical robots playing soccer. The selection of this game as a laboratory task was motivated by the fact that the realization of this complicated multidisciplinary task is very difficult. The entire game can be divided into a number of partial tasks (evaluation of visual information, image processing, hardware and software implementation of distributed control system, hard-wired or wireless data transmission, information processing, strategy planning and controlling of robots). The task is a matter of interest for both students and teachers, and allows direct evaluation and comparison of various approaches. For the improvement of game strategy, we are developing an abstract description of the game and propose ways to use this description (e.g. for learning rules and adapting team strategies to every single opponent).

We are building upon our previous work - the hardware implementation and basic control of robots - and we would like to achieve a higher level control of complex game strategies.

The rest of the paper is organized as follows: First, we briefly describe the base hard-ware and software implementation. Then, we describe the representation of the game field using abstract grids. After that, we describe possible game strategies. Using the abstract grids and game strategies, we explain how to learn rules that describe specific game strategies. Particular attention is paid to the learning how to use latent semantic analysis. We conclude with the discussion of the presented approach.

2 Technical Implementation

Embedded systems are represented by two teams (own and opponent) made up of up to 11 autonomous mobile robots. The core of an embedded control system is a digital signal processor Motorola - DSP56F805. PWM output of the signal processor is connected to a pair of power H-bridge circuits, which supply a pair of DC drives with integrated pulse encoders. For communication, the communication module is used with the control IC Nordic nRF2401, which ensures communication with a higher level of the control system.

The higher level control system is represented by a personal computer. At the PC input, a signal that represents a picture of a scene with robots scanned with CCD aerial camera is entered. At the output, a radio line that transmits commands for all own mobile robots is connected.

The software part of a distributed control system is realized by decision making and executive agents. The algorithm of agents cooperation was proposed with the control agent on a higher level. The algorithms for agents realized in robots are the same. The control agent determines the required behavior of the whole control system as the response to the dynamic behavior of robots. One's own global strategy in the task and knowledge of prior situations saved in the database of the scene can also be determined. The agent on a higher level controls the other agents [1].

Another task is the transformation that converts the digital picture into the object coordinates (robots and ball in the task of robot soccer) saved in the database of the scene [2]. This database is common for all agents in the control system. Used agents structure was described in [3]. Each agent sees the entire scene and is capable of controlling its own behavior. The basic characteristic of a control algorithm of a subordinate agent is the independence on the number of decision making agents for robots on the playground.

Both teams (one's own and the opponent's) have a common goal, to score a goal and not to allow any goals from opponent. For successful assertion of one's own game strategy the extraction and knowledge of an opponent's game strategy is very important. Strategy extraction algorithms are created from the opponent's game strategy database and from object coordinates of the picture scene.

3 Game Field Description

The game system can be described as up to twice eleven autonomous mobile robots (own and opponent's robots), which are situated on a field measuring 280x220cm. In

our approach we are using a software simulator for this robot soccer game [4]. Robots create a very dynamic environment. This environment is scanned by a CCD aerial camera with a sample frequency (in the present time) up to 50 fps.

The neuronal net of a control agent in a sensation module process the picture signal and encoded information (position, orientation) is saved in one of output vectors of scene database [5]. Scene database is common for all agents. Both agent teams have a common goal to score the goal and not to get any goal. For a success, it is also advisable to extract the strategy of the opponent team. The extraction and knowledge of opponent game strategy is an approach that is known to be successful in other situations as well [6].

Our approach of game representation is based on separation of game into logical and physical part. Logical part includes selection of strategy, calculation of robot's movements, and adaptation of rules to opponent's strategy. Physical part contains real movement of robots on game field and recognition of opponent's turn. Logical part is independent on physical part, because we can calculate moves of opponent's robots as well as moves of our robots. Next advantage of separation is that logical part is independent on size of game field and resolution of used camera. In logical part, the game is represented as *abstract grid* with very high resolution, which guaranteed very precise position specification of robots and ball. But this very detailed representation of game field is not suitable for strategy description, because that brings requirements of many rules for description of behavior of robots. Therefore, for strategy description is used so-called *strategy grid* with much less resolution than abstract grid. This simplification of reality is sufficient, because it is not necessary to know exact position, but only approximate position, of robots for strategy realization (Figure 1). When physical part is used, then we must only transform coordinates from abstract grid into coordinates based on game field size and camera resolution.

Fig. 1. Inner game representation

4 Game Strategy

The game strategy can be dynamically changed based on the game progress (i.e. the history and the current position of the players and the ball [7]). The game progress can be divided in time into the following three ground playing classes (GPC). GPC of game opening (GPCO), GPC of movements in game site (GPCS), GPC of game end (GPCE). The game progress, especially in the GPCS class, can be also divided into the following two game playing situations (GPS):

- GPS of attack (GPSA). The interactions of simple behaviours cause the robots to fall into a V-formation where the ball is in motion roughly towards the opponents goal.
- GPS of defence (GPSD). When the ball is not moving roughly towards the opponents goal, the robots move around it to form an effective barrier and to be in a good position for recovery.

Each GPC has its own movement rules. The classes GPCO and GPCE consist of finite number of possible movements that are determined by initial positions of players and the ball. The class GPCS has virtually unlimited number of possible movements. The movements are determined by the current game situation (GPS) and by the appropriate global game strategy (in next GGS). The movement of the particular robot is determined by the current game class and situation, and also by the robot role. For example, the goalkeeper's task is to prevent the opponent to score a goal. His movements are in most cases limited along the goalmouth near of goal line. The preferred movements are in goal line direction. The preference of these movements comes from the particular GGS, where the goalkeeper prevents to score a goal in the way of moving in the position between the central goal point and the ball (or the expected ball position). The preference of other movement directions is created using GPSA, where movements of goalkeeper secure kicking the ball from the defense zone.

It should be noted that above presented categorization of game play progress is rather a tool for analysis and training and description of initial team habits than strict differentiation of game situations. During the dynamic strategy adaption, the rules change and should evolve toward more suitable opponent anti-strategy. So the content of each GPC is changing in time and the particular strategies influence each other and also the strategies belonging to other classes.

5 Basic Description of Strategy Selection Process

In this section we describe our approach for learning game strategy from observation. Our goal is to learn an abstract strategy. The main steps of the learning process are:

- Transformation of observations into abstract grids,
- Transformation of observations into strategy grids,
- Learning a strategy based on the observed transitions in the strategy grid.

We adopt definition of strategy [8]: „Strategy is the direction and scope of an organization over the long-term: which achieves advantage for the organization through

its configuration of resources within a challenging environment...“ Strategy application for one movement of players is computed in following steps:

1. Get coordinates of players and ball from camera.
2. Convert coordinates of players into strategic grid.
3. Convert ball and opponents' positions into abstract and strategic grids.
4. Choose goalkeeper and attacker, exclude them from strategy and calculate their exact positions.
5. Detect strategic rule from opponents' and ball positions.
6. Convert movement from strategic grid to physical coordinates.
7. Send movement coordinates to robots.

Each strategy is stored in one file and currently consists of about 15 basic rules. Furthermore the file contains metadata Information about the name of strategy, the algorithm to strategy choosing, the author responsible for current strategy, the date of last modification, the size of strategic grid, strategic rules.

Each strategic rule consists of five records. The rule ID and description (e.g. Rule 1 "Attack1"), the coordinates of our players in strategic grid (e.g. .Mine a6 c7 d6 e3 f9), the coordinates of opponent's players in strategic or abstract grid (e.g. .Opponent d3 e7 e8 g2 k6), the ball coordinates in abstract or strategic grid (e.g. .Ball i6), strategic or abstract grid positions of the move (e.g. .Move a6 g7 f5 j3 i8). From observation of opponent's strategy a new set of rules can be written, without necessity of program code modification. Furthermore, there is a possibility of automatic strategy (movement) extraction from running game.

There exist two main criteria in the Rule selection process. The selection depends on opponents' coordinates, mines' coordinates and ball position. The strategy file contains rules, describing three possible formations suggesting danger of current game situation. The opponent's team could be in offensive, neutral or defensive formations. Furthermore, we need to weigh up the ball position risk. Generally, opponent is not dangerous if the ball is near his goal. The chosen rule has minimal strategic grid distance from current.

Optimal movements of our robots are calculated by applying minimal distance from strategic grid position. The goalkeeper and attacking player, whose distance is closest to the ball are excluded from strategic movement and their new position is calculated in exact coordinates. To summarize, the strategy management can be described in the following way:

- Based on incoming data from the vision system, calculate abstract and strategy grid coordinates of the players and the ball,
- The abstract grid is then used to decide which player has under the ball control,
- This player is issued a kick to command that means that it has to try to kick the ball to a given strategy grid coordinates,
- All other players are given (imprecise) go to coordinates. These coordinates are determined by the current game strategy and are determined for each robot individually,
- The goalkeeper is excluded from this process since its job is specialized, and does not directly depend on the current game strategy.

6 Future Research

The need to learn opponent's strategy from the game and decide an appropriate counter-strategy in response was identified in the previous section. Also, verification of the created strategy is of notable importance. An off-line (out of the gameplay) verification process validates the strategy and ensures that there are no:

- Contradictory rules leading to contradictory game situations
- Extra rules made immediately or in more steps the same game situations

Such a verified game strategy can improve complex goal-targeted robot behavior in practice. We aim to extend our framework with strategy validating and optimizing component based on genetic algorithms.

Genetic Algorithms are powerful and popular optimization and search algorithms inspired by natural evolution introduced by John Holland and extended by David Goldberg. GA is wide applied and highly successful variant of evolutionary computation [9]. GA operates over a population of potential solutions encoded into chromosomes. Each chromosome is rewarded with a fitness value expressing its suitability as a solution of given problem. The workflow of GA consists of iterative application of genetic operators on population of chromosomes (Figure 2). Genetic operators are:

- Selection operator: to select the fittest chromosomes from the population to be parents. Through this operator, selection pressure is applied in the population.
- Crossover operator: for varying chromosomes from one population to the next by exchanging one or more of their subparts.
- Mutation operator: random perturbation in chromosome structure; used for changing chromosomes randomly and introducing new genetic material into the population.

Fig. 2. Evolutionary algorithm

A population of chromosomes, in particular iteration of evolutionary computation, is called a generation. When applying evolutionary optimization to a robot soccer game, a strategy can be seen as a generation of rules encoded as binary chromosomes. The evolution over strategy rules should lead to a new improved set of rules that will

form a better counter-strategy against a particular opponent. The challenge of this approach is to find suitable rule encoding (that will allow easy application of genetic operators and respect the nature of the investigated problem) and discover a useful fitness function to compare rules.

The introduced approach to robot soccer strategy is based on the overall knowledge of global game situations. Next, robot movements are computed after an analysis of overall positions of all players and ball. In fact, robots are being moved by an omniscient operator and not moving independently. More advanced multi-agent approaches also incorporate agent behaviors, such as an ant algorithm, a stigmergy algorithm, a multi-agent coordination and control using techniques inspired by the behavior of social insects. In such a game, the robot players will resolve simple or trivial game-play situations according to their own decisions. This should mirror the instinctive behavior of robotic agents and local knowledge following their individual, simple behavior patterns. A pattern example can be as follows:

- The attacker moves towards the ball (when the ball is moving to opponents goal)
- The attacker possessing the ball moves towards the competitors' goal
- The defender moves between the ball and own goal area (when the ball is moving towards own goal).
- The defender moves quickly between the ball and its own goal, when the opponent attacks and there are no defenders in position.
- The goalie moves towards the ball in an effort to perform a kick-off when the ball is too close to the goal area. Otherwise, he guards the goal.

The strategy concept presented in previous sections should then be used to resolve complex non-trivial game situations (like standard situations in a real soccer game) or to incorporate surprising, innovative moves to the game. If robot activity is partly or mostly independent, there will be no need to evaluate the global game situation and search for appropriate moves in every GPS. The gained processor time can be used for improved strategy learning, game strategy optimization, and optimized game strategy applied in certain gameplay situations (the opponents' team loses the ball when attacked by more players; own team gains advantage by attacking with more players, posing a greater threat to opponents goal). This can lead to notably superior results.

Supplementary out of the play analyses of game history, recorded by the means of abstract grid and strategy rules, are used for fine tuning robot behavior as described above and for development of strategy rules fitting accurately to a particular opponent.

7 Conclusion

The main goal of the control system is to enable an immediate response in real time. The system response should be sooner than time between two frames from the camera. When the time response of the algorithm exceeds this difference the control quality deteriorates. The method we described provides fast control. This is achieved by using rules that are fast to process. We have described a method of game representation and a method of learning game strategies from observed movements of players.

The movements can be observed from the opponent's behaviour or from the human player's behaviour. We believe that the possibility of learning the game strategy that leads to a fast control is critical for success of robotic soccer players. Like in chess playing programs, the database of game strategies along with the indication of their success can be stored in the database and can be used for subsequent matches.

In the future, we want to use modular Q-learning architecture [10]. This architecture was used to solve the action selection problem which specifically selects the robot that needs the least time to kick the ball and assign this task to it. The concept of the coupled agent was used to resolve a conflict in action selection among robots.

Acknowledgement

The Grant Agency of Czech Academy of Science supplied the results of the project No. p. 1ET101940418 with subvention.

References

1. Horák, B., Obitko, M., Smid, J., Snášel, V., Communication in Robotic Soccer Game. *Communications in Computing* (2004) 295-301.
2. Holland, O., Melhuish, C., Stigmergy, self-organisation, and sorting in collective robotics. *Artificial Life*, (2000) 173-202.
3. Srovnal, V., Horák, B., Bernatik, R., Strategy extraction for mobile embedded control systems apply the multi-agent technology. *Lecture Notes in Computer Science*, Vol. 3038. Springer-Verlag, Berlin Heidelberg New York (2004) 631-637.
4. FIRA robot soccer, <http://www.fira.net/>
5. Berry, M. W., Browne, M., *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM Book Series: Software, Environments, and Tools (1999)
6. Slywotzky, A.J., Morrison, D., Moser, T., Mundt, K., Quella, J., *Profit Patterns: 30 ways to anticipate and profit from strategic forces reshaping your business* (1999)
7. Veloso, M. and Stone, P., Individual and collaborative Behaviours in a Team of Homogeneous Robotic Soccer Agents, *Proceedings of International Conference on Multi-Agent Systems* (1998) 309-316.
8. Johnson, G., Scholes, K., *Exploring Corporate Strategy: Text and Cases*. FT Prentice Hall, (2001)
9. Mitchell, M., *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA (1996)
10. Park, K.H., Kim, Y.J., Kim, J.H., Modular Q-learning based multi-agent cooperation for robot soccer, *Robotics and Autonomous Systems*, Elsevier, 35 (2001) 109-122.