Control and Monitoring of a Robotic Soccer Team: The Base Station Application

Nuno M. Figueiredo, António J. R. Neves and Nuno Lau

Institute of Electronics and Telematics Engineering of Aveiro
University of Aveiro, 3810-193 Aveiro, Portugal
{nuno.figueiredo,an,nunolau}@ua.pt

Abstract. In robotic soccer, teams of autonomous robots play soccer according to rules similar to the official FIFA rules. The game is refereed by a human and his orders are communicated to the teams using an application called "Referee Box". No human interference is allowed during the games except for removing malfunctioning robots and re-entering robots in the game. The base station, a software application as described in this paper, has a determinant role during the development of a robotic soccer team and also during a game. This application must control the agents interpreting and sending high level instructions, like `Start` or `Stop`, and monitor information of the robots, for example the position and velocity, allowing easily to attest the feasibility of the robots behavior. This paper discusses the importance of the control and monitoring of a robotic soccer team, presenting the main challenges and the approaches that were used by the CAMBADA team in the conception of the base station application. As far as we know, no previous work has been published about the study of these important problems and the discussion of an efficient architecture to a base station application. The results obtained by the team confirms the good performance of this software, both during the games and in the development of the team.

1 Introduction

Robotic soccer is nowadays a popular research domain in the area of multi-robot systems. RoboCup[1] is an international joint project to promote research in artificial intelligence, robotics and related fields. RoboCup chose soccer as the main problem aiming at innovations to be applied for socially relevant problems. It includes several competition leagues, each one with a specific emphasis, some only at software level, others at both hardware and software, with single or multiple agents, cooperative and competitive.

In the context of RoboCup, the Middle Size League (MSL) is one of the most challenging. In this league, each team is composed of up to 6 robots with a maximum size of $50cm \times 50cm$ width, $80cm$ height and a maximum weight of 40Kg, playing in a field of $18m \times 12m$. The rules of the game are similar to the

[1] http://www.robocup.org/

official FIFA rules, with minor changes required to adapt them for the playing robots [1].

The rules of this league establish several constraints to simplify perception and world modeling. In particular, the ball is orange, the field is green, the field lines are white and the players are black. The duration of a game is 30 minutes, not including a half-time interval of 5 minutes. The game is refereed by a human and his orders are communicated to the teams using an application called "Referee Box" operated by an assistant referee.

No human interference is allowed during the games except for removing malfunctioning robots and re-entering robots in the game. Each robot is autonomous and has its own sensorial means. They can communicate among each other and with an external computer through a wireless network. This external computer, that has no sensor of any kind, runs the base station application. The base station only "knows" what is reported by the playing robots and the orders received from the referee box. The agents should be able to evaluate the state of the world and take decisions suitable to fulfill the cooperative team objective.

CAMBADA[2], *Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture*, is the Middle Size League Robotic Soccer team from the University of Aveiro. The CAMBADA research project started in 2003, coordinated by the Transverse Activity on Intelligent Robotics (ATRI)[3] group of the Institute of Electronic and Telematic Engineering of Aveiro (IEETA)[4]. Since then, it has involved people working on several areas for building the mechanical structure of the robot, its hardware architecture and controllers [2] and the software development in areas such as image analysis and processing [3–7], sensor and information fusion [8], reasoning and control [9].

Since its creation, the team has participated in several competitions, both national and international. Each year, new challenges are presented, and new objectives are defined, always with a better team performance in sight. After achieving the first place in the national competition Robótica 2007 and Robótica 2008, the 5th place in the world championship RoboCup 2007, in 2008 the team achieved the first place in the world championship RoboCup 2008.

This paper focuses the role of the control and monitoring activities in the development and during a game of a robotic soccer team. We start this paper clarifying the concepts of control and monitoring in the context of this paper. This step takes special importance due to the fact of these concepts are widely used in other contexts and could led to a misunderstanding about our purposes. In Section 3, we present an overview of the communication and internal architecture of the CAMBADA team, in order to explain some of the used approaches. In the Section 4 will be addressed the main challenges and requirements that these activities demands on to the development of the application. Section 5 describes the approaches used in the development specifically for each activity

[2] http://www.ieeta.pt/atri/cambada

[3] http://www.ieeta.pt/atri

[4] http://www.ieeta.pt

focusing the implemented features. Finally we present our main conclusions and future improvements to the project.

2 Control and monitoring in a robotic soccer team

For control and monitoring we can find multiple definitions of their meaning. In the context of this paper, to avoid some misunderstandings, we will clarify what we are meaning when we refer each one.

We can find the “Control” concept applied in many areas like engineering or management. In engineering the control concept implies an ability to keep a certain observed variable (output) constrained, acting in the variable/variables (inputs) that change the state of the referred observed variable. In a robotic team the control of the robots is autonomous, is the robot that have to observe the state of the world (outputs) and act accordingly (inputs). However in the testing phase, and only in that, is useful provide a certain level of control in order to directly observe some wanted behavior.

In this paper when we refer control, it means the ability to instruct the robotic agent to perform a certain high level behavior, like start, stop or assign a certain role. However it is important to refer that during the game the control of the robots are autonomous and guaranteed by individual processing of the information of the game.

The monitoring of the robots implies the ability to observe a specific behavior or characteristic. In the context of this paper, performing monitoring is essentially concerned with three things:

- observe the consequences of the control actions;
- observe the internal states and hardware information of the robots;
- detect errors related with incorrect algorithm implementations.

3 Communication and internal architecture

In this section, the communication and internal architecture of the CAMBADA team will be explained. The description will be based on the characteristics of interest that were used in the development of the base station application. More specifically, concerning communication, we present the way as information is exchanged between the different agents, what are the used protocols and access mechanisms implemented in the team.

Concerning the internal architecture of the team, it will be explained the structures that support the information. This description will be based essentially in the description of the `WorldState` model implemented.

To exchange the information between the agents in the field, it is used a Wireless LAN [12, 13] and each team has a different one. In the connection between Referee Box and base station is used a wired TCP LAN connection using a standard protocol [1] to report the events of the game. In the team network no restrictions are made, each team is free to choose the connection protocol and the information exchanged between its agents. In Fig. 1 are presented a model of the connections used.

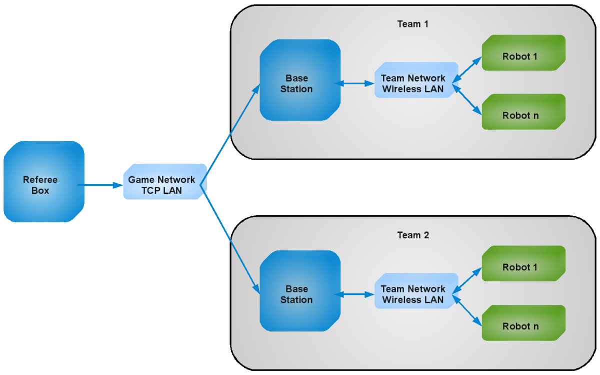


Fig. 1. The network configuration used in a soccer game.

The internal communication protocol is based on the exchange of the `WorldState` information between each agent. At each communication cycle the agents report their internal state creating a virtual representation, shared by all, of the global `WorldState`. This virtual representation is named Real-Time DataBase (RTDB) [10]. This mechanism simplifies the information management and access. In the point of view of an application, one single data structure represents all the needed information of the shared `WorldState`. The base station application uses this mechanism to access the individual information of each player and also to send information to each one. In Fig. 2 are shown how information is exchanged between all the agents in the field.

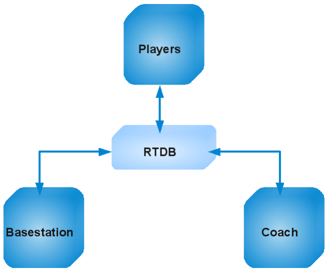


Fig. 2. The communication between all robots is accomplished through the RTDB.

Regarding the internal architecture, we will detail the most relevant information exchanged in this process, more specifically, what are the most relevant information that can be useful to the mentioned activities, control and monitoring.

As it was mentioned before, each agent exchange information of their `WorldState` among all the team mates. For this purpose, we categorized the information accordingly with the mentioned activities:

- **Control Information** – this information are related with the high level orders that can be sent to the agents, during a game or in the development phase of the robots:
  - game signals, including `Start` and `Stop` actions;
  - team color;
  - goal side;
  - role.
- **Monitoring Information** – this kind of information is related with the internal states and the functional information that indicates the performance of the agent in a certain instant:
  - game states;
  - position of the ball and the robot;
  - angle of the robot;
  - velocity of the ball and the robot;
  - batteries information;
  - ball engaged and visible;
  - the actual Role and Behavior;
  - the actual Team color and goal side;
  - debug point.

The base station uses these information to perform the control and monitoring of the robots.

4 Main challenges and requirements

The base station application must provide a set of tools to perform the activities mentioned above. Regarding the control activity, this application must allow high level control of the robots sending basic commands/information in particular the run and stop commands, play mode, role assignment, etc.

This application must also provide a high level monitoring of the robots internal states, namely the position in the field, velocity, battery charge, among other relevant information related with the robots and the game.

Furthermore, this application should provide an easy mechanism that can be used to easily show a specific behavior of the robot, allowing debugging.

Besides that, the base station has a fundamental role during a game, while receiving the commands from the referee box, translating them to internal game states and broadcasting the results to the robots. During a game, no human interference is allowed except for removing malfunctioning robots and re-entering robots in the game.

The role of the base station during these phases, development and games, demands the fulfillment of some requirements, being the most important the following:

Reliability / Stability: during the game, the base station is not accessible for human interaction of any kind and thus, it has to be a very robust application, all team depends on that.

Usability: the information displayed in the base station should be easy to interpret, allowing, for instance, a fast detection of a problem in a robot. It should be possible to choose different levels of details in the displayed information. Moreover, the base station has to be easy to use, allowing an intuitive management of the robots.

Adaptability: a robotic soccer team is in a permanent development stage, which may lead to significant changes within a short period of time.

Specifically to each phase the base station should provide the following features:

- **Development phase**

  Manual role assignment: acting as a cooperative team, each robot has a specific role which is, during a real game, dynamically assigned. In the development phase, it should be possible to manually assign a role to a specific robot.

  Local referee box: the base station should provide an interface widget to emulate a real referee box in order to simulate events of a real game.

  Visualization Tool: the application should provide a representation of the field and the robots in that context. Moreover, some visual information should be attached in order to improve the visual perception of the internal states of each robot.

  Multi-windows solution: the application should be a multi-window environment, allowing the user to choose between different levels of information. At least, three different levels of information should be provided: a work level that presents the controls of the robots and basic status information; a visual level that presents visual information of the position of the robots and, finally a detailed level that shows all the information related to the robots.

  Adaptable windows geometry: the multi-windows system should adapt to monitors with different resolutions. According to the new MSL rules, the base stations of each team must use an external monitor provided by the organizing committee.

- **Game situation**

  Robust communication skills: the correct operation of the team during the game is fully dependent on the communication between the robots, the base station and the referee box. Hence, the base station should provide a robust communication layer.

Automatic processing of the game states: the base station should process the commands received from the referee box allowing the robots to change their internal game states accordingly. One specific action should be the changing of the field side at half time.

In the next section it will be detailed the approaches followed in the conception of the CAMBADA base station in order to fulfill these requirements.

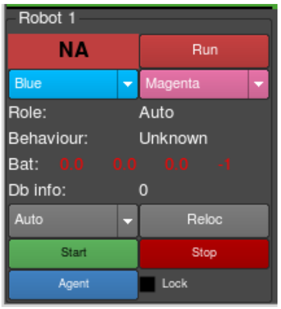
5 The CAMBADA base station approach

Regarding the challenges and requirements mentioned in the previous section, we will detail the used approaches in the conception of the base station application. We have divided, once again, the description in the mentioned activities in order to more precisely describe the features implemented for each one.

5.1 Performing control

Merging the available methods provided by the communication protocol of the team we were able to implement a widget that allows an high level control of the robots. In the Fig. 3 are shown the possible actions that can be used to control the robots. All the actions were grouped to each robot and are accessible in a delimited space in order to improve the usability. These actions represents the enable/disable of each robot, the team color and goal color (in spite of the current rules don't specify goal colors, we decide keep it in order to facilitate the monitoring process), the role of the player, the re-location button, the start and stop that controls remotely the software in each robot and the bottom to launch remotely the agent.

Additionally, were created two other widgets, one to control all the team and one that implements a local referee box. These two widgets are shown in Fig. 4.



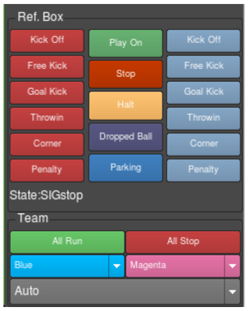
Role: Auto

Behaviour: Unknown

Bat: 0.0 0.0 0.0 -1

Db info: 0

Fig. 3. Available controls and basic information of one robot.



State:SIGstop

Fig. 4. Local Referee Box and team controls.

5.2 Monitoring

Regarding the monitoring activity, we developed a visualization widget that makes a representation of the field and the robots. This visualization widget shows the robots number, position and orientation and the ball that each robot view. Additionally was implemented a mechanism that allows to change the orientation of the field, in order to turn possible to monitor the robots in any position of the field, increasing the usability. In Fig. 5 are shown the visualization widget and this described mechanism.

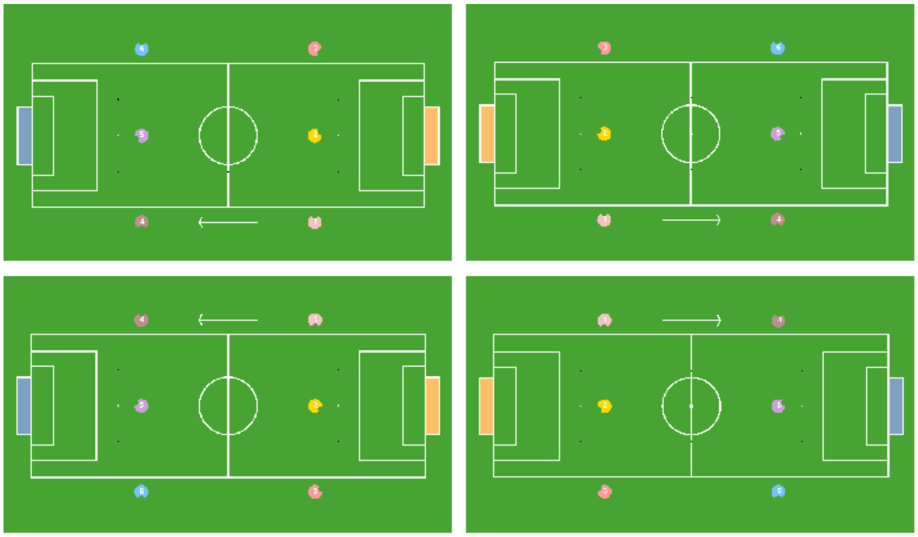
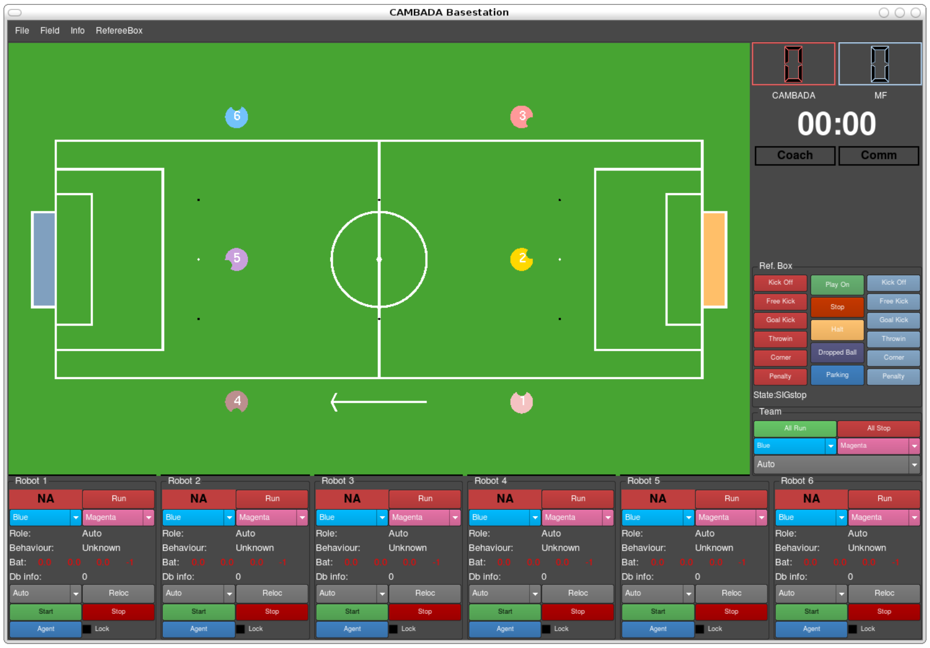


Fig. 5. Different views of the field. This mechanism allows the developers to adjust the field orientation accordingly to their point of view of the field

The base station has three separated windows representing three different levels of information. The main level shows the controls and relevant information about the robots state, other window only shows the visualization widget (this is the window to monitor the game, according with the new rules) and finally we implemented a window with full information of the robots, all the information available in the RTDB is shown in this window. In Fig. 6, 7 and 8 are shown these three windows.

In order to perform debugging in the development phase, it was implemented, in the visualization widget, a debug mechanism. Is possible to enable this mechanism writing in a specific field of the RTDB. This field is a vector with two dimensions representing a position on the game field. There are one point of debug *per* robot and if enabled in the base station this point can be shown in the game field together with the representation of the robots. This point is free



State:SIGstop

Robot 1
Role: Auto
Behaviour: Unknown
Db info: 0

Robot 2
Role: Auto
Behaviour: Unknown
Db info: 0

Robot 3
Role: Auto
Behaviour: Unknown
Db info: 0

Robot 4
Role: Auto
Behaviour: Unknown
Db info: 0

Robot 5
Role: Auto
Behaviour: Unknown
Db info: 0

Robot 6
Role: Auto
Behaviour: Unknown
Db info: 0

Fig. 6. The base station `Main Window`.

to use and can represent whatever the developer wants. In Fig. 9 are shown this debug mechanism.

Additionally, the third window, considered as the full information window, allows to perform debug to the robot state, more precisely in the transition between the roles and behaviors states.

All the project were developed using the Qt [11] library using a modular architecture. This increased the reliability and stability allowing to test each module before the integration in the project.

6 Conclusion and Future Improvements

This paper presents the base station application of the CAMBADA team under the perspective of a tool that can enhance the control and monitoring capabilities of the team developers. During the paper, we reason about the importance of these activities in the phases of development and during the games, highlighting the main challenges and requirements that this application presents. Furthermore, we presented some main approaches used to solve the challenges that were identified showing some features implemented.

The current version of the base station was tested in the Portuguese championship Robótica 2008 and in the world championship RoboCup 2008, showing

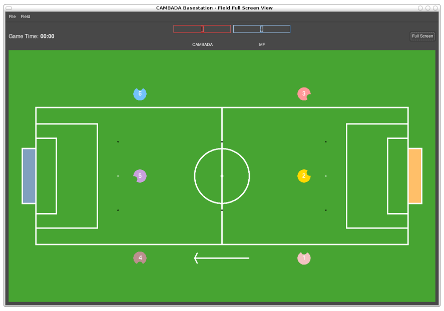
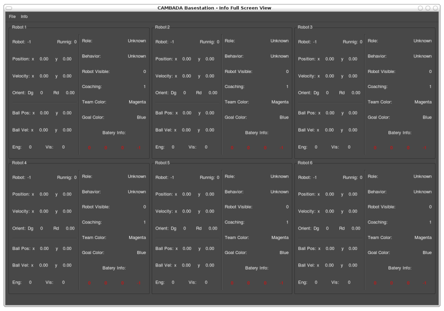


Fig. 7. The base station `Field View Window`.



Robot 1
Robot: -1 Runnig: 0
Position: x 0.00 y 0.00
Velocity: x 0.00 y 0.00
Orient: Dg 0 Rd 0.00
Ball Pos: x 0.00 y 0.00
Ball Vel: x 0.00 y 0.00
Eng: 0 Vis: 0
Role: Unknown
Behavior: Unknown
Robot Visible: 0
Coaching: 1
Team Color: Magenta
Goal Color: Blue
Batery Info:
0 0 0 -1

Robot 2
Robot: -1 Runnig: 0
Position: x 0.00 y 0.00
Velocity: x 0.00 y 0.00
Orient: Dg 0 Rd 0.00
Ball Pos: x 0.00 y 0.00
Ball Vel: x 0.00 y 0.00
Eng: 0 Vis: 0
Role: Unknown
Behavior: Unknown
Robot Visible: 0
Coaching: 1
Team Color: Magenta
Goal Color: Blue
Batery Info:
0 0 0 -1

Robot 3
Robot: -1 Runnig: 0
Position: x 0.00 y 0.00
Velocity: x 0.00 y 0.00
Orient: Dg 0 Rd 0.00
Ball Pos: x 0.00 y 0.00
Ball Vel: x 0.00 y 0.00
Eng: 0 Vis: 0
Role: Unknown
Behavior: Unknown
Robot Visible: 0
Coaching: 1
Team Color: Magenta
Goal Color: Blue
Batery Info:
0 0 0 -1

Robot 4
Robot: -1 Runnig: 0
Position: x 0.00 y 0.00
Velocity: x 0.00 y 0.00
Orient: Dg 0 Rd 0.00
Ball Pos: x 0.00 y 0.00
Ball Vel: x 0.00 y 0.00
Eng: 0 Vis: 0
Role: Unknown
Behavior: Unknown
Robot Visible: 0
Coaching: 1
Team Color: Magenta
Goal Color: Blue
Batery Info:
0 0 0 -1

Robot 5
Robot: -1 Runnig: 0
Position: x 0.00 y 0.00
Velocity: x 0.00 y 0.00
Orient: Dg 0 Rd 0.00
Ball Pos: x 0.00 y 0.00
Ball Vel: x 0.00 y 0.00
Eng: 0 Vis: 0
Role: Unknown
Behavior: Unknown
Robot Visible: 0
Coaching: 1
Team Color: Magenta
Goal Color: Blue
Batery Info:
0 0 0 -1

Robot 6
Robot: -1 Runnig: 0
Position: x 0.00 y 0.00
Velocity: x 0.00 y 0.00
Orient: Dg 0 Rd 0.00
Ball Pos: x 0.00 y 0.00
Ball Vel: x 0.00 y 0.00
Eng: 0 Vis: 0
Role: Unknown
Behavior: Unknown
Robot Visible: 0
Coaching: 1
Team Color: Magenta
Goal Color: Blue
Batery Info:
0 0 0 -1

Fig. 8. The base station `Info Window`.

a high level of reliability and stability helping the team to achieve the successful results of the last year.

Besides of these promising results, some work still has to be done, namely future improvements in the development of new control capabilities allowing a low level control performed by this application individually to each robot. This new type of control will allow us to act directly on the main features of the robots, like motor velocity, orientation, kick power, etc. To enhance the monitoring capabilities, we are planning to implement, in a first phase, new visual debugging points, not only a numerical vector but also binary states (like semaphores). In a second phase, we are planning to implement a reliable positioning debugging based on image processing where static cameras will be used in the field with the aim to provide reference information about the objects in the field (robots, balls, obstacle, etc). This information will be used to validate the algorithms of positioning of the team.

Also in monitoring, we verified some problems with the quantity of the information displayed. It is planed in the future some improvements regarding a new way of presenting the information in order to optimize the available space.

7 Acknowledgment

This work was supported in part by the FCT project PTDC/EIA/70695/2006.

References

1. MSL Technical Committee 1997-2009, “Middle Size Robot League Rules and Regulations for 2009”, 2008.
2. J. L. Azevedo, B. Cunha, and L. Almeida, “Hierarchical distributed architectures for autonomous mobile robots: a case study”, in *Proc. of the 12th IEEE Conference on Emerging Technologies and Factory Automation, ETFA2007*, Greece, 2007, pp. 973–980.

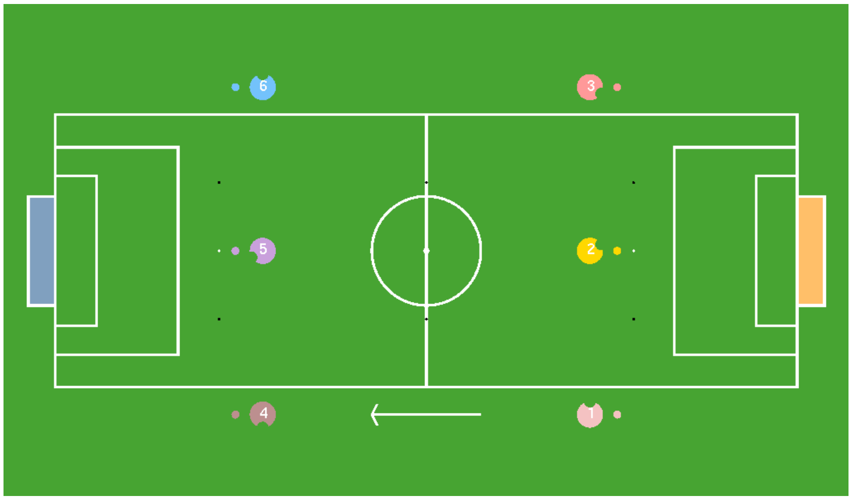


Fig. 9. The 6 robots in the field and the respective debug point.

3. A. J. R. Neves, G. Corrente, and A. J. Pinho, "An omnidirectional vision system for soccer robots", in *Proc. of the EPIA 2007*. 2007, vol. 4874 of *Lecture Notes in Artificial Inteligence*, pp. 499–507, Springer.
4. A. J. R. Neves, D. A. Martins, and A. J. Pinho, "A hybrid vision system for soccer robots using radial search lines", in *Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, Portuguese Robotics Open - ROBOTICA'2008*, Aveiro, Portugal, april 2008, pp. 51–55.
5. D. A. Martins, A. J. R. Neves, and A. J. Pinho, "Real-time generic ball recognition in RoboCup domain", in *Proc. of the 11th edition of the Ibero-American Conference on Artificial Intelligence, IBERAMIA 2008, IROBOT Workshop*, Lisbon, Portugal, october 2008, pp. 37–48.
6. P. M. R. Caleiro, A. J. R. Neves, and A. J. Pinho, "Color-spaces and color segmentation for real-time object recognition in robotic applications", *Revista do DETUA*, vol. 4, no. 8, pp. 940–945, June 2007.
7. B. Cunha, J. L. Azevedo, N. Lau, and L. Almeida, "Obtaining the inverse distance map from a non-svp hyperbolic catadioptric robotic vision system", in *Proc. of the RoboCup 2007*, Atlanta, USA, 2007.
8. J. Silva, N. Lau, J. Rodrigues, and J. A. Azevedo, "Ball sensor fusion and ball interception behaviours for a robotic soccer team", in *Proc. of the 11th edition of the Ibero-American Conference on Artificial Intelligence, IBERAMIA 2008, IROBOT Workshop*, Lisbon, Portugal, october 2008, pp. 25–36.
9. L. S. Lopes N. Lau and G. A. Corrente, "CAMBADA: Information sharing and team coordination", in *Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, Portuguese Robotics Open - ROBOTICA'2008*, Aveiro, Portugal, april 2008, pp. 27–32.
10. L. Almeida, F. Santos, T. Facchinetti, P. Pedreira, V. Silva, and L. S. Lopes, "Coordinating distributed autonomous agents with a real-time database: The CAMBADA project", in *Proc. of the 19th International Symposium on Computer and Informa-*

tion Sciences, ISCIS 2004. 2004, vol. 3280 of *Lecture Notes in Computer Science*, pp. 878–886, Springer.

11. Trolltech Co, “Qt cross-platform application framework”.
URL: *http://www.trolltech.com*
12. F. Santos, L. Almeida, P. Pedreiras, L.S. Lopes, and T. Facchinetti, “An Adaptive TDMA Protocol for Soft Real-Time Wireless Communication among Mobile Autonomous Agents”, in *Proc. of the Int. Workshop on Architecture for Cooperative Embedded Real-Time Systems, WACERTS 2004*, 2004, pp. 657–665.
13. F. Santos, G. Corrente, L. Almeida, N. Lau, and L.S. Lopes, “Selfconfiguration of an Adaptive TDMA wireless communication protocol for teams of mobile robots”, in *Proc. of the 13th Portuguese Conference on Artificial Intelligence, EPIA 2007*, 2007.