ISePorto Robotic Soccer Team for Robocup 2003

J. M. Almeida, A. Martins, E. P. Silva, J. Baptista, A. Patacho, L. Lima, V. Cerqueira, C. Almeida, R. Picas

LSA -Autonomous Systems Lab
Instituto Superior de Engenharia do Porto
R. Dr. António Bernardino de Almeida 431
4200-072 Porto, Portugal
{jma, amartins, eaps, jpaulo}@dee.isep.ipp.pt, AntonioPatacho@eseig.ipp.pt

Abstract. This paper describes the design and implementation status of the Vision, Localisation and Control systems of the ISePorto robotic football team for participation in Robocup 2003 Middle Size League. The objectives guiding the project were the applications and research in sensor fusion, hybrid control and coordination systems. The system has also an educational support role. A special attention is made to the custom design to allow the execution of complex manoeuvres and team coordinated behaviours. The robot has different pass, shot, and manoeuvre capabilities providing high level tactical and strategic planing and coordination.

1 Introduction

The ISEP Autonomous Systems Lab. (LSA) robotic football team provides an excellent tool to develop and demonstrate the research in the areas of interest associated with autonomous systems. These are mainly sensor fusion, mobile robotics navigation, nonlinear hybrid feedback control and coordination. Additionally to the research interests, the laboratory has a strong educational purpose, being the robot team a good support to curricular and extra-curricular work in the areas of mechatronics, electronics and embedded systems.

The remaining of the paper overviews the robot design, the vision system, the control and navigation issues, coordination and strategy guidelines finishing with the current team status.

2 Robot Design and Development System

The team robot was designed and implemented from scratch in order to be a suitable testbed for advanced multirobot coordinated control. The mechanical design has taken in account the requirements to execute complex manoeuvres and therefore not posing harsh limits on the control, navigation, or coordination developments. This led us to the development of an innovative rotational kicker and to use two pan controlled cameras.

Whence, the robot has three parts: a circular mobile base, a kicker connected to a structure that rotates around a central vertical axle, and on top, a computational an electronics module fixed relatively to the base with a pan mounted camera. The system is mechanically modular; it can be used in different configurations, such as different kicker designs with the same base. The base contains two differential traction 24 V DC motors with optical encoders for motor control and vehicle odometry, two 12V lead acid batteries with 5Ah or two 12V NiMH packs with 6Ah, the kicker rotation motor and the motor power drives boards. The kicker uses a DC motor and mechanical spring with a camber and was designed to allow different kick strengths. This, coupled with the kicker rotation allows the robot to perform complex manoeuvres.

The main computational system consists of a 5.25" SBC (ICP NOVA7896FW with a 900Mhz Celeron) and IDE 24Mb Flash disk. Each robot communicates with the team and the host visualisation computer by an Ethernet wireless modem compliant with IEEE 802.11b.

The motor control is made in a custom designed multi-axis control board, comprising a FPGA (FLEX 10K10) and a dedicated microcontroller (T89RD2). It communicates with the main CPU through a PC104 connector (ISA bus). The PC104 form factor reduces size and it is a reliable connection system. The board implements the PID controllers for 4 axis at 2KHz (traction, kicker and head camera rotation), receiving encoder information and providing a sign magnitude PWM control signal to the power drives. Additionally, this board interfaces an optical switch to calibrate the angle of the rotating kicker and head camera with respect to the robot base, and several digital I/O lines used for power supply control like turning on/off the power to the motors. The robot has also a custom developed ring of IR range measuring sensors for short distance obstacle detection (up to 0.8m).

3 Development System

The onboard embedded computer runs a Linux operating system from the flash disk. For development convenience, we use a root filesystem over NFS.

A specific distribution was developed for the vehicles and a standard Linux kernel (2.4.19) was modified in order to provide some real-time functionalities and additional development tool support. These modifications include the following patches: high resolution clocks, pre-emptive kernel and Linux Trace Toolkit support.

The software design follows a modular and hierarchic multi-threaded software architecture [3]. We use the Posix soft real time extensions: a priority pre-emptive scheduler and all the application memory resources are prevented from swapping to disk with mlock. The robots use a developed software API for communication by IP datagrams in a broadcast or multicast configuration.

There have been developed graphical tools for visualisation of robot state and perceived sensor information, camera parameters and colour segmentation adjustment, and user interface for robot operation. In addition, a remote variable inspector and logging tool is being developed.

4 Vision System

The vision system uses two USB cameras (Philips PVC740K) placed in a new housing that interface a CS-mount wide angular lens (a 2.8mm, F1.4, maxvision MVL2810M). One of them mounted on a head central pan unit and used mainly for long range vision and localisation proposes and the other fixed to the kicker used mainly for fine ball control, obstacle perception and goal free area identification. Both cameras acquire images at 30fps, the head camera with a resolution of 320x240 and the kicker camera with a resolution 160x120. The image processing threads are capable to process all the acquired images. This processing is mainly done in a colour-segmented image.

The colour segmentation algorithm uses a pre-determined set of regions in the YUV in order to detect the game relevant colours. The colour segmentation algorithm implementation follows a very efficient method proposed in [10]. This process is still very sensitive to lighting conditions. Some problems with overlapping regions are noticed in the cases of colour that form a cluster diagonal to the YUV limit box. In order to improve the segmentation for colours within a diagonal cluster this is bounded by a set of boxes in YUV. All this process of colour segmentation can be very difficult to do without a good tool. Additionally, the problem can be even harder if we add all the camera parameters (shutter speed, gain, white balance: red and blue, brightness, etc). In practice, we have a well-established set of colour limits and normally adjust the camera parameters until have a good segmentation. After that, we can adjust some colour limits to further improve the colour segmentation.

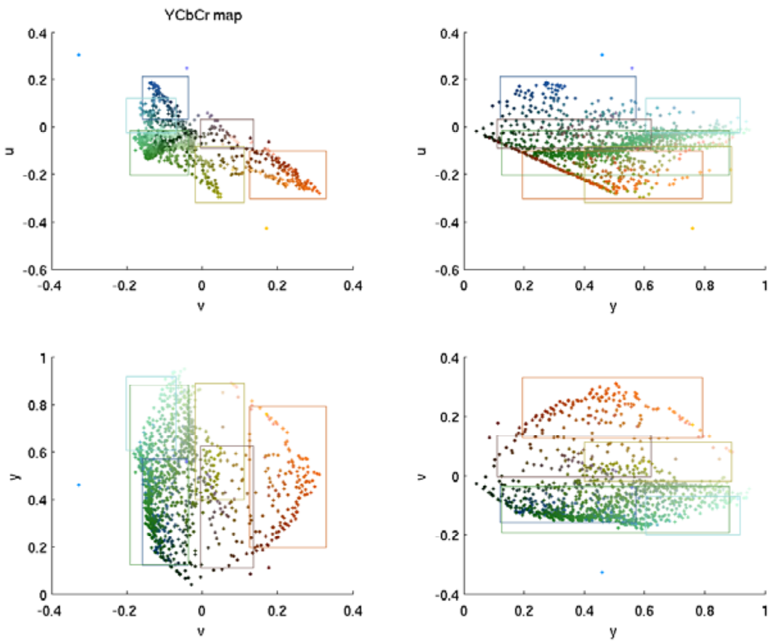

Fig. 1. Example of a YUV Colour map and segmentation colours limits, visualised by the colour segmentation graphical tool

The main characteristic of our vision system is the ability to perform a fast global analysis of the all image in order to decide where to conduct a more detailed one. An additional improvement is accomplished with the integration of landmark predictions in the grabbed image or where the camera should point.

The landmarks and world objects perceived by the vision system are the ball, Corner flags, Goals (Limits and Areas), Team colour marks and dark Obstacles. The ball is detected using a blob-based algorithm. This provides information regarding ball centroid and bounding box for all possible clusters. The obstacles are identified by some threshold on the percentage of black, blue and yellow on each of the defined partitions in the image (shown in Fig. 2, in the top, right image) induced by a set of pie like partitions of the camera 2D plane referential. The camera perspective and image distortion parameters are used to compute those image partitions.

The other objects are detected along specified scan lines. A hierarchical approach is taken to eliminate false detection. Due to wide angular characteristics of the lens used (needed for convenient field of view) the image received has a high distortion. This distortion exists in both cameras but is more relevant on the kicker camera.

The distortion is taken in account in the scan lines. For example: a horizontal line (see the white line in the top of the Fig. 2) is computed in the distorted space, and the result is used in the scan. The information obtained in the distorted space is then converted to a corrected image coordinates and then to camera referential. The advantage of the use of this scan line is twofold. First, the angle to some target obtained this way becomes less sensitive to camera oscillations. Second, this scan allows a fast search for target candidates with the guaranty that if they are in the field of view they must intercept the horizontal plane where the search is conducted.

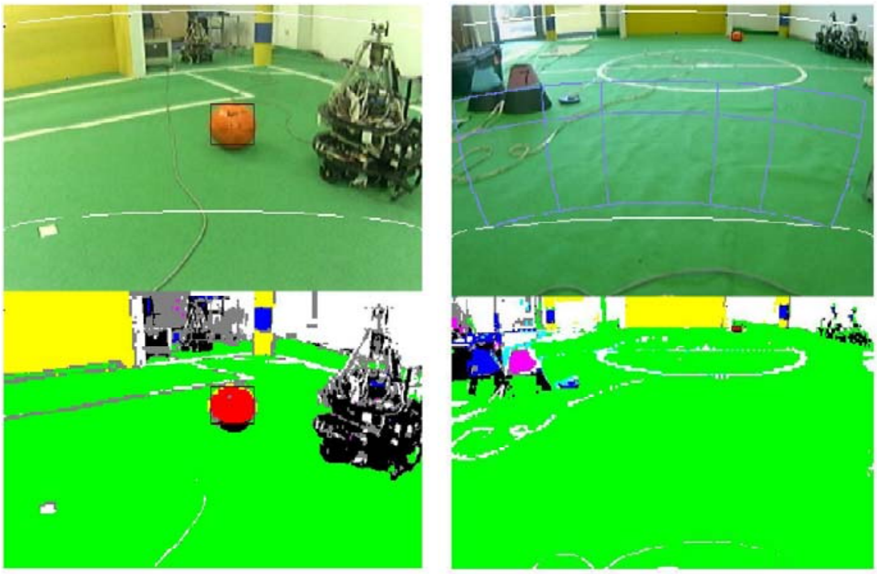

Fig. 2. Original and colour segmented images. Over the images we can see the Horizontal scan line in the original images (head camera, marked in white on top image), the Ball limits represented by a black box around the ball, and the Image partition for obstacles detection represented by a blue grid on the right top image.

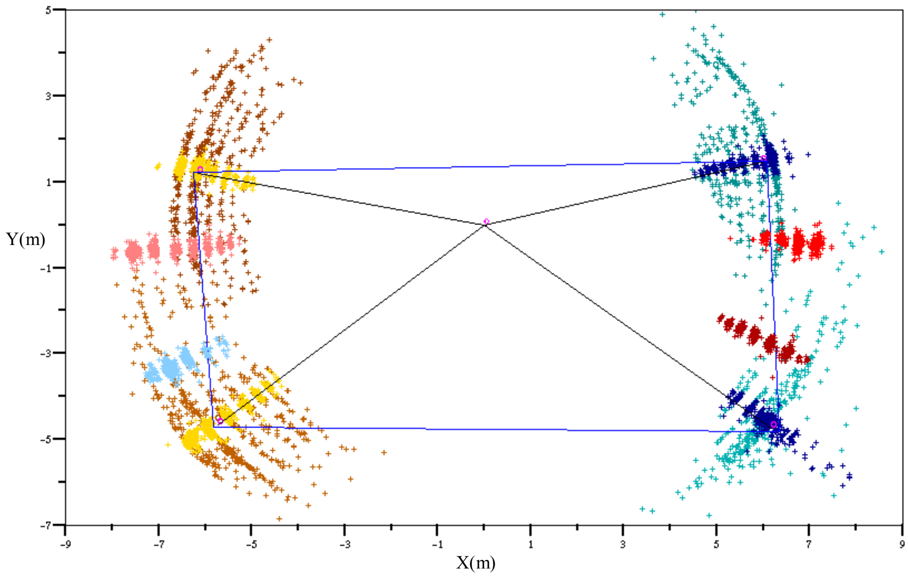

Fig. 3. Corner and goal vision data measures from the head-rotating camera

In fig. 3, we can see a 2D representation of the raw data measures of the angular and distance of the 4 corner flags, and two goals. For the corner flags, we can see the result of using the angular velocity and an estimated delay in the image acquisition to correct the corner angular measures. The corrected measures are drawn in yellow and blue over the raw corners angle data drawn in brown and cyan. In the next table, are shown some statistics for the all the corners.

Table 1. Statistics for corner measures

Angular Statistics (deg)	E(θ)	STD(θ)	E(θ_corr)	STD(θ_corr)	E(θ_w0)	STD(θ_w0)	**Distance (m)**	E(d)	STD(d)	E(d_w0)	STD (d_wo)
BYB1	13.94	8.85	13.65	0.81	13.55	0.22	**BYB1**	6.13	0.38	6.26	0.17
BYB2	-36.48	7.59	-37.45	0.87	-37.12	0.55	**BYB2**	7.79	1.03	7.96	1.03
YBY1	169.01	11.26	169.00	1.04	168.95	0.43	**YBY1**	6.26	0.40	6.36	0.28
YBY2	218.48	9.07	219.08	0.82	219.06	0.34	**YBY2**	7.38	0.67	7.49	0.66

It can be observed in the previous table the statistics for three scenarios. Each line corresponds to a different corner (BYB: blue-yellow-blue corners; YBY: yellow-blue-yellow corners). The situations analysed for the same data are, the raw measurements statistics (E(θ), STD(θ)), the subset of measures taken with null camera angular velocity (E(θ_w0), STD(θ_w0)) and the overall corrected (E(θ_corr), STD(θ_corr)) estimates taking in account the camera angular velocity and delay in the image capture (when we perform a acquisition command in video4linux, the image that we will receive can be more than 80ms old).

Although the standard deviation of the θ_corr is about 10 times less than the raw θ standard deviation, it's still 2 to 4 times greater than the standard deviation of the angle measures taken with the still camera. This is due mainly to some uncertainty in the acquisition time of the image.

5 Localisation

In the Robocup to achieve good team play capabilities, is required a good robot localisation and control. Besides the robot self-localisation it is also necessary to perform adequate ball (and if possible other players) localisation.

The ball position estimation uses a Kalman filter taking in account the robot model and the information received from the two cameras from the vision system. Adaptive ball measurement algorithms are used for different cases of ball position and occlusion in the image. An updated ball position and velocities estimate is maintained in the robot internal state.

The self-localisation is mainly done by the fusion of vision measures of world landmarks (goals, corner and ground marks), internal sensors (odometric and magnetic compass) and external vision measures from other robots, using Kalman filtering methods, using a classic vehicle and observation model. Here the key point, concerns to data association phase, were corner and goal edge measures are associated with the position information in the world model.

In the next figure, we can see that the filter converges even with a small initial position error (about one meter in both axes). The innovation sequence falling within the 3σ innovation covariance bounds. The theta related innovation appears to be zero-mean, but the distance one has a small bias.

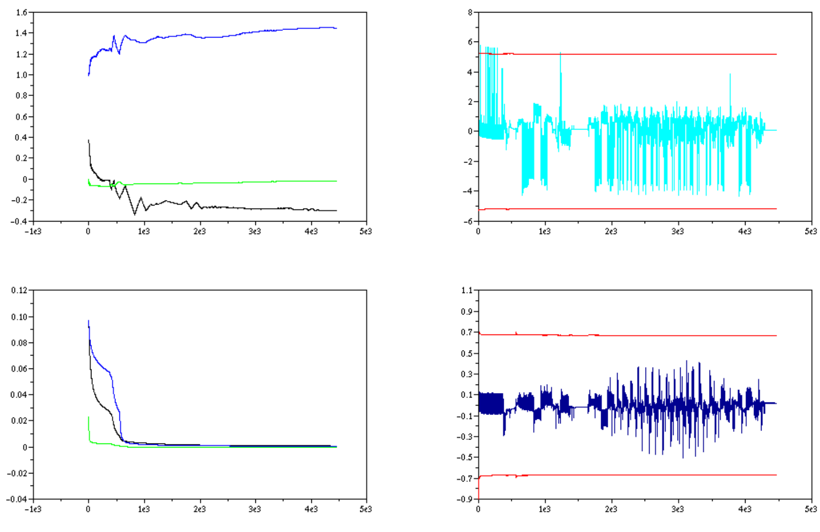

Fig. 4. Robot State (upper left; x: black, y: blue, theta: green), Diagonal of the covariance matrix (down left), distance Innovation component (upper right) and theta innovation component (down-right)

For filter initialisation or re-initialisation, the robot performs a scan of the world in order to create a map that are matched against the world model in order to select the valid and invalid map objects. Then the valid ones are used by the initialisation algorithm. We plan to test fusion algorithms based in covariance intersection [5], for filter initialisation/re-initialisation proposes using the difference between the two angles to two landmarks, since those measures are more insensitive to the camera oscillations and image acquisition delays, than distance or single landmarks angles.

Furthermore, each robot has a world state with some knowledge of position, attitude and his derivatives, with some uncertainty measure for all the game moving objects. This is accomplished with distributed sensor fusion, where the vision sensors play a key role in sensing. Also, a distributed dynamic camera allocation and managing is under study. In the presence of communications problems, the desired information must be perceived individually by each player. In the last case, in spite of an obvious degradation of the world model, some team coordination must be accomplished by the perceived information of our robots.

6 Control

The motion control architecture approach [1] is based in atomic parameterised hybrid feedback controller, also known as manoeuvre. These controllers incorporate both continuous and event driven feedback. This approach involves the atomic parameterised hybrid controllers synthesis. A set of manoeuvres solving specific classes of motion problems is defined and implemented. These are classified according to the patterns of the associated constraints and objectives. Those manoeuvres are the resources to the coordination level.

The sets of manoeuvres that are being synthesised are:

Motion without ball:

- Move to location avoiding obstacles
- Block goal path
- Approach ball with defined attitude
- Move to maximise target vision information
- Goalkeeper block goal path

Coordinated robot and kicker motion:

- Smooth ball reception
- *Ball guidance:*
  - With robot rotation around ball centre
  - Straight line
  - Curve
  - With directional shoot and defined strength
  - Interception and kick
  - Lateral pass with kicker rotation

As an example in the next figure a state diagram is presented for a simple attack manoeuvre.

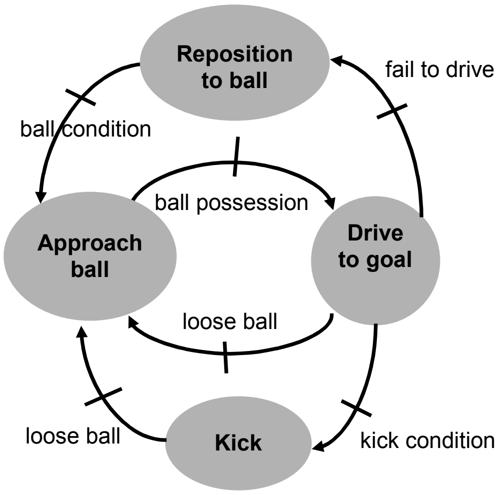

Fig. 5. Simple attack manoeuvre

In this case, the attack manoeuvre has four discrete states: approach ball, reposition to ball, drive to goal and kick. Each one corresponds to a continuous control mode.

This hybrid automata (see [7], [8] and [9] for a formal treatment on hybrid systems) uses game specific events related with the robot, ball and target goal, as triggers for the discrete state jumps (more precisely the guard conditions detects these events and the reset relations are the identity). The robot initially approaches the ball, and when has control of it, drives the ball towards the target goal. The first control law provides the approaching movement and the second defines the angular velocity and orientation of the kicker in order to preserve ball control and drive it to target. In the event of losing the ball, the robot executes a different control mode to re-approach the ball from a direction compatible with the target ball. If the robot has ball control and some kicking condition is met (minimum distance and good goal probability of success) it kicks the ball.

7 Coordination Strategy

The team game evolution is coordinated in structural way [2], [4], by defining the tactical function (goal keeper; defence; middle field; attack) for each robot .The robot adopts the corresponding tactical policy accordingly to its perception about the companions in the field. The player has one main place in the overall strategy (as consequence of its tactical position), but this can be reconfigured dynamically. Only one robot has the possibility to adopt the goal keeper position.

The overall strategy solution results from the composition of the decisions taken in a distributed way by the operational robots in the field.

Several types of abstract game information are considered in the strategy decisions. A key aspect in the decision process is the perceived ball state in terms of possession or reachability. This state can be:

- *Ball unknown* – Robot does not know the ball state
- *Ball ours* – Our team has ball control
- *Ball theirs* – Opposite team has ball control
- *Ball near* – Ball within reach by the robot (and consequently by the team)
- *Ball far* – Ball far from current robot

Each robot has an observer responsible for ball state determination. This is achieved using local information and information received by the team mates (e.g if one robot does not see the ball it can determine its state if other informs of ball possession or localisation).

The analysis of the vectors: game phase, ball possession and current topology formation in conjunction with the current robot game role and its perceived topological position determines the coordination level for each robot. An evaluation of the next action to be taken is made by the maximisation of hypothesis success.

The coordination level is implemented by a modular and distributed controller synthesis through the composition of discrete observers (corresponding to the analysis vectors) and a discrete controller parameterised by the adopted tactical functionality
There is a high level automata for the vehicle coordination that depicts the discrete game status. The robot can be either off game, or in several game and pre-game situations. These range from the positioning phase in the beginning before game, the waiting to start (already positioned in the field) and a general playing phase. It can also be temporarily suspended and return to game (for instance in a substitution or repair). Each game phase can be itself a complex hybrid automata with different discrete states. The hierarchical structure of the control architecture allows the modularity and incremental development of the system. In addition provides the execution of complex behaviours and refinement in the overall coordination strategy.

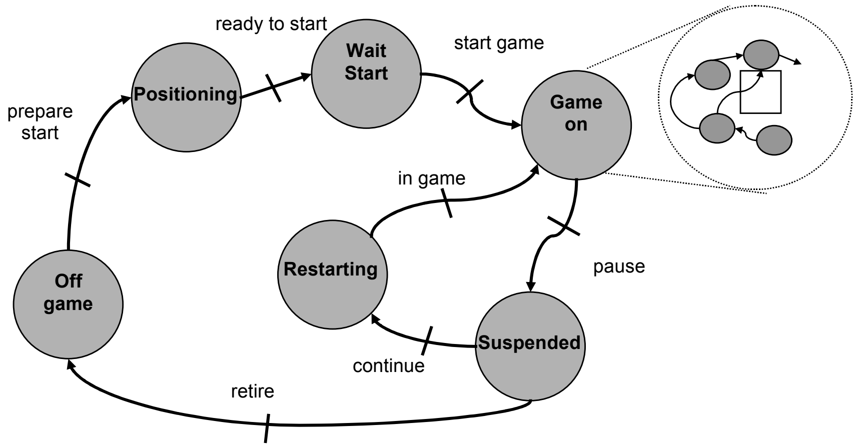

Fig. 6. Coordination high level automata

8 Conclusions and Future Work

The design of the vision, control and localisation subsystems of the robotic soccer team ISePorto is presented. We have a full team plus a spare robot ready. The team has participated in several competitions namely in the German Open 2002 and 2003, Robótica 2002 in Aveiro, Portugal, in the 1st 2003 National tournament between the Portuguese teams held in ISEP, and Robótica 2003 in Lisbon, Portugal with good results.

This complex mobile robotics is a motivating example to the research and development of control and navigation. The control and navigation architecture for the robots is described. This hierarchic architecture entails the use of hybrid control automata in order to achieve complex behaviours. The coordination and game strategy issues are also referred.

The team status is still at an initial development stage, with a vast number of areas to be developed. In these we could name the other players distributed localisation, game status observer development, team topological formation classifier, additional control manoeuvres and game control/API software development.

Acknowledgement

The authors would like to thank the LSA team for their efforts and dedication in developing the robots and support on the participation of competitions and events. This work is sponsored by IPP, ISEP, Salvador Caetano and Bosch Security systems.

References

1. J. M. Almeida, F. L. Pereira, J. B. Sousa, "A Hybrid Feedback Control System for a Nonholonomic Car-like Vehicle," IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, 1997.
2. E. P.Silva, F. L. Pereira and J. B. Sousa, "On the Design of a Control Architecture for an Autonomous Mobile Robot," chap. 15, in Advances In Intelligent Autonomous Systems, ed. S.G. Tzafestas, Kluwer Academic Publishers, 1999.
3. J. M. Almeida, E. P. Silva, A. Martins, et al, "Functional Layer of the control architecture PO-ROBOT Project Final Report, 1999.
4. E. P. Silva, F. L. Pereira, J. M. Almeida, "Coordination and Organisation layer", PO-ROBOT Project Final Report, 1999.
5. Covariance Intersection Working Group (CIWG), "A Culminating Advance in the Theory and Practice of Data Fusion, Filtering, and Decentralised Estimation".
6. A. Martins, J. Almeida, E. P. Silva and J. P. Baptista, " Design of the ISePorto Robocup Middle-Size League Robotic Soccer Team: Control, Localisation and Coordination ," Proc. MED2002 Mediterranean Conference On Control and Automation, Lisbon, July , 2002.
7. J. Lygeros, C. Tomlin and S. Sastry, "Controllers for reachability specifications for hybrid systems,", Automatica, Vol, 35, pp 349-370, 1999.
8. R. K. Boel, B. De Schutter et al, "Approaches to Modelling, Analysis, and Control of Hybrid Systems, ", Technical Report BDS:99-04, Delft University of Technology, 1999.

9.] Thomas Henzinger, “The Theory of Hybrid Automata”, Technical Report UCB/ERL M96/28, U. C. Berkeley, 1996.
10. J. Bruce, T. Balch and M. Veloso, “Fast and Inexpensive Color Image Segmentation for Interactive Robots,” Proc. 2000 IEEE/RSJ International Conf. On Intelligent Robots and Systems, vol 3, pp2061-2066, October, 2000