FROM THEORY TO PRACTICE: THE FURGBOL TEAM

Silvia Botelho*, Emanuel Estrada*, Gisele Simas*, Humberto Souto Jr*, Lorenzo Taddei*, Marcelo Galeriano*, Mariane Medeiros*, Paulo Drews Jr*, Rafael Deon*, Renan Almeida*, Renato Neves*, Thiago Santos*

**Fundação Universidade Federal do Rio Grande - FURG*
Av. Itália Km 8, Rio Grande, RS, Brazil

Emails: `silviacb@ee.furg.br`, `maninho@ee.furg.br`, `gisele@ee.furg.br`, `humberto@ee.furg.br`, `lorenzo@ee.furg.br`, `galeriano@ee.furg.br`, `mariane@ee.furg.br`, `paulo@ee.furg.br`, `rafaeldeon@ee.furg.br`, `renan@ee.furg.br`, `renato@ee.furg.br`, `thiagoss@ee.furg.br`

Abstract— The present paper describes FURGBOL, a brazilian low cost F-180 team of autonomous robots, its implementation and our experiences with it. We propose an architecture composed by three main stages: i.A Deliberative Stage, ii A Communication Stage and, iii. A Embedded Reactive Control. It also describes the relevant aspects of our architecture, like software, hardware and design issues.

Keywords— Autonomous Robot Soccer, Computional Vision, Artificial Intelligence, Embedded Reactive Control.

Resumo— O presente artigo descreve o modelo do time brasileiro de futebol de robôs autonômos FURGBOL, categoria F-180, desenvolvido visando baixo custo, sua implementação e as experiências obtidas. Nós propomos uma arquitetura composta por três estágios principais: i.Estágio Deliberativo, ii. Estágio de Comunicação e, iii. Controle Embarcado Reativo. Também é descrito os aspectos relevantes de nossa arquitetura, como ocomplexidade de software, o hardware e as versões do projeto.

Palavras-chave— Robôs de Futebol Autonômos, Visão Computacional, Inteligência Artificial, Controle Embarcado.

1 Introduction

The field of application of multi-robots systems has evolved in the last few years. (Parker et al., 2005). RoboCup (Shimizu and Nagahashi, 2005; Egorova et al., 2003; D'Andrea, 2003; Loomis et al., 2003; Kuth et al., 1999) is a long-term effort of the academic and industrial research community to develop teams of robotic football/soccer players. Fast movement and team coordination must be achieved at the same time, and this is a major challenge. The robocup scenario is being used as an excellent test-bed for research in several areas associated with Computer Engineering and Science. Several domains must be treated, for instance:

- capable single players (vision, real-time sensor fusion and control, autonomous agents, robotics)
- teamwork (multiagent cooperation, context recognition)
- understanding the competition (cognitive modeling) the ability to develop and execute plays and strategies in real time (strategy acquisition, real-time reasoning and planning and reactive behavior)

To resolve all the items listed above is a research challenge. Several important approaches propose to build sophisticated multi-robot teams through the combination of expensive and complex hardware and mechanical devices (Shimizu and Nagahashi, 2005; Egorova et al., 2003; D'Andrea, 2003; Loomis et al., 2003). From an educational perspective, the RoboCup Competitions is also a great motivation for exposing students to design, build, manage, and maintain complex robotic systems. However, nowadays, how to participate in a RoboCup Competition with a very limited budget, bringing together recent state-of-art robotic concepts? Is it possible to implement good solutions and sophisticated design methodologies with low cost robotic and sensors plataforms? The leap from theory to robotic implementation is often difficult to do, and to do well or efficiently, even more difficult.

The FURGBOL F-180 Team is an effort of the Department of Computer Engineering of the Fundação Universidade Federal do Rio Grande, Brazil. Our goal is to stimulate research, teaching, and applications in the fields of artificial intelligence and collaborative robotics. FURGBOL team is composed by a group of undergraduated. Our team use inexpensive and easily extendible hardware components and a standard software environment. Besides, the FURGBOL platform is entirely based on open source software. Even with a very limited budget (U$ 1500,00), FURGBOL has show to be a relatively successful approach; since it started, in 2001, we are three times champion of Brazilian Robocup and vice-champion of Latin American Robocup twice.

This paper describes a set of inexpensive issues associated with our F-180 Robocup Team. In section 2, we introduce our architecture composed by three main stages: Embedded Reactive Control, Communication and Deliberative Stages. Next sections detail each one of these stages. Finally, we present our implemented system which illustrates the principal aspects of our contribution.

2 An Overview of Our Team

The idea is to have an omnidirectional team to play soccer. Our robots uses omni directional wheels, and each wheel has its own motor. In this way each motor needs an independent control and imposes a force in one from the two possible directions. The resulting force composed by the forces (from each wheel) moves the robot towards the desired direction.

Resulting from the need of several reconstruction issues, we decided to rebuild the chassis and wheels. For that reason, everything described here is currently in development and might differ from the final version being used in the competition.

Chassis The chassis is used to hold all other components and also works as heatsink for the voltage converter. It consists of one laser-cut aluminium plate. The robot chassis has a diameter of 161mm and a height of 100mm. See figure **??**.

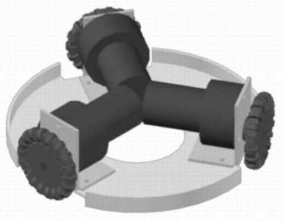

Figure 1: The CAD chassis of FURGBOL robot.

Architecture Starting from the **Plan-Merging** Paradigm for coordinated resource utilization - and the **M+ Negotiation for Task Allocation - M+NTA** for distributed task allocation, we have developed a generic architecture for multi-robot cooperation (Botelho and Alami, 2000). This architecture is based on a combination of a local individual reactive control and a central coordinated decision for incremental plan adaptation to the multi-robot context. In this paper we present an adaptation of this system to use in a RoboCup Team, see Figure 2.

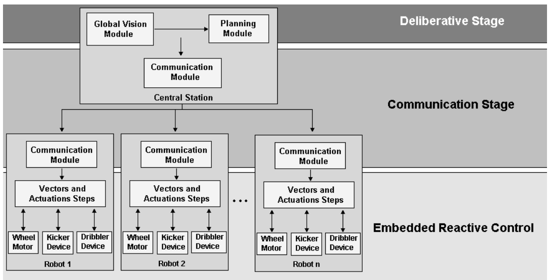

Figure 2: Our architecture with three main Stages.

A Centralized Deliberative System is in charge of the global perception of the field and teams, identifying robots and the ball; planning trajectory and a desired behavior of each robot. The communication system exchanges information between robots and Central Station (CS). Finally, we have a reactive embedded control. This stage receives the high level global information from CS, reacting to local environmental changes. Next Sections detail each one of the architecture stages.

3 A Deliberative Central Stage

We assume that robots and ball are agents. A state machine is associated with each agent. A Central Deliberative System perceives the environment (agent states) and plans actions and tasks associated with each member team. Thus, this stage has two main modules: the Global Vision Module and the Planning Module.

3.1 The Global Vision Module

This system works with a set of frames captured by n firewire video cameras located above the field. These frames are combined to provide an internal world model to other software components. Therefore simultaneous threads read, convert, process and provide the picture data in a modular vision framework. New features are currently implemented to improve the quality of the world model data.

The vision system identifies robots and the ball giving their position and velocities through a set of image processing techniques implemented by the system; at first, a correction of radial distortion is performed. The next step is a segmentation based on the HSV color space analysis. Finally, a set of heuristics are used to localize each agent in the scene.

The Correction of Radial Distortion In the literature, several techniques treat radial and tangential distortions caused by lens at the image. As the radial component causes larger distortion, most of the work concerns this problem (Bailey, 2002). The two following equations 1 are used to correct radial distortion.

$$x_u = x_d + k_1 x_d \left(x_{d^2} + y_{d^2} \right), y_u = y_d + k_1 y_d \left(x_{d^2} + y_{d^2} \right) \quad (1)$$

where (x_u, y_u) are the corrected coordinates of the distorted point measure (x_d, y_d), and k_1 is the first term of the radial correction series, truncated at its quadratic term. Giving a frame with n pixel this algorithm has complexity $O(n)$.

The HSV Color Transformations and Automatic Color Calibration Initially, the captured images are in RGB format (Red Green Blue). Aiming a more robust system in relation to luminosity variation, a HSV (Hue Saturation Value) color space transformation is performed (Gonzalez and Woods, 2001). In Robocup Rules, only a limited number of colors, c, can be used to identify the ball and robots. In accordance with the official rules, these colors are: blue and yellow for the teams identification; orange for the ball; light green, light pink and cyan for the robots identification; green for the field; and white for the field marks. As each one of these colors determines a specific color class, the purpose of this module is to classify each pixel of the frame belonging to one of these classes.

So, the process starts with a offline automatic calibration step where the H, S and V intervals are defined for each class through the k-means clustering algorithm. The k-means hands out information into partitions (clusters) by similarity of characteristics. This is made in a automatic process, more robust and eficient than manually calibration. In images, the pixels are distributed in K classes by an interactive method that stops when the precision of the algorithm has been reached out. Inicial values for H, S and V componentes are suplied for each class, creating its centroids. So, the euclidian distance of each pixel to each class is calculated and the pixels are related to the closest class found. Based on the pixels groups produced in the last step, new centroids are calculated and all process is repeated. At the end of execution, the k-means algorithm returns the central values of each one of the k classes, with diferent values of H, S and V components. The k-means has a time complexity that is dominated by the product of the number of pixels n, the number of clusters c, and the number of iterations.

After this, all the image is classified, pixel by pixel, through the H, S and V components (Bruce et al., 2000). In adittion, the analisis of some characteristics of colors in RGB format is performed, in a process that complements the classification in HSV format. This analisis avoids the ambiguousness in classification of some colors as orange and light pink.

Segmentation / Localization Step The object segmentation is based in formation of circular blobs - a number of adjacent pixels of same color class. To assure the correctioness of segmentation and avoid interferences caused by capture noises, this region must be bigger than a predefined minimum size. At first, the position of all blobs found at one image are stored. So, it is started a process of connection among the blobs considering the two basic color classes (blue and yellow - colors of the teams) and the diameter of robots. For each blue or yellow blob there are some secondary blobs around (cyan, light pink and light green) used to provide the identification and orientation of robots. All secondary blobs localizated inside of the boundaries of robots are conected to them and the remaining blobs are discarded. Finally, the orange blobs undergoes an analysis that determines the position of the ball into the field. This process has complexity $O(n^2)$. Notice that there are two potencially problems in searching for orange blobs: *i.* even if the calibration has been well carried through, this color easily can be confused with other colors, like the light pink; and *ii.* sometimes, robots hide the ball in a way that no ball is found into the field. For these reasons, orange blobs inside a determined radius around the blue and yellow blobs are discarded, using a $O(m \log m)$ complexity algorithm, where m is the number of blobs. In this case the last position of the ball from previous frame will be maintained (Loomis et al., 2003).

State	description	Actions
Disputed	two adversary robots are near to the ball	to retrieve the adversary ball
Ownership	only team members are near to the ball	to move (with ball)
Free	nobody near to the ball	to move (without ball)
Adversary	only opposite robots are near to the ball	to follow an adversary
Goal	the ball is into goal limits	to kicker

Table 1: The ball states and Robot Actions.

With the current and past positions, Planning Module plans the actions associated with the team members.

3.2 The Planning Module

The planning module is based on a world model which models the state of each agent in the game. We use a set of state machines whose nodes are related to the state of the players and ball and the transitions are given in function of the dynamics of the game.

A Perception Step This step transforms position and velocity information into states associated with each agent. This step has complexity $O(r^2)$ where r is the number of team robots. A set of states and transitions (actions) were defined. See table 1 for ball states, and a set of actions (transitions). They are defined based on the relative positions between robots and ball.

The robots and ball will assume topological labels, called areas, that identify their localization inside the field: for x axis (that joins both goals) they might be either in defense areas, halfway or attack; for y axis (perpendicular to the x): left side line, right side line and halfway.

A Role Assignment Step With the ball state and topological labels already defined, the Planning Module calculates a set of actions to be achieved by each team member. For that, three kinds of roles are defined: the goal-keeper, the defense and the attack. Each role has a own state machine and a different strategy to move and dribble (see Figure 3 and 4).

Figure 3(a) shows the goal-keeper role. This strategy calculates an intersection between a vertical straight line (parallel to the middle line) that cross its center and a straight line gotten from the current and previous position of the ball. The intersection point is where the robot must move itself.

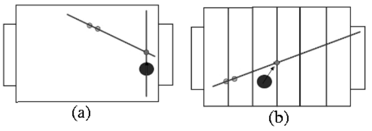

Figure 3: (a) Goal-Keeper Strategy (b) Defense Strategy

The defense role uses as information the topological label of the ball, to move the robot to defense the goal, see Figure 3(a). Each area has a vertical straight line, wich represents its center. In Figure 3(b) we can see a center line of the area besides the current ball area (in direction to adversary goal). This strategy calculates the intersection between this red line and an straight line created from the current and previous position of the ball. The intersection point is to where a defense robot must go.

The attack role is divided in three different strategies, according to ball and robots localization.

The first one is applied when the nearest robot to the ball can attack. It activates the dribbler device and go ahead loading the ball or kick it into the opposite goal direction. It happens when a straight line from the center of the ball and the center of this robot cross the opposite goal vertical straight line resulting in a point inside of the goal limits.

The second attack strategy occurs when the previously described intersection results in a point out of the limits of the opposite goal. In this case the robot must be posed in a valid position so that it can load the ball to the goal. This position can be gotten by the intersection of a straight line that joins the adversary goal and the ball centers with a circumference of pre-defined radius centered at the ball.

The last approach happens when the intersection of the first approaching strategy results in a point out of the limits of the opposite goal and the ball is not between the robot and the opposite goal. In this case, an intersection between a vertical straight line that pass in the center of the ball and a circumference with a pre-defined radius centered at the same one is carried through. After, the second and first strategies of approach will be applied.

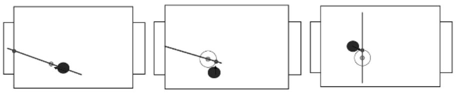

Figure 4: First, second and third attack strategies, respectively

A Trajectory Planning Step Each one of these three basic roles supplies a target position to where each robot must move itself. We use an *approximated cell decomposition method* to achieve each individual target position. This method has complexity $O(p.o)$ where p is the number of pixels and o is the number of obstacles. This approach allows a planning of the robot trajectory ensuring that it will not collide. This method was chosen for being simple, to supply a uniform decomposition and to be suitable for make possible the attainment of accuracy (resolution) (Amato, 2004).

The method of approximated cell decomposition as shown by (Latombe, 1991) divides the field in three possible cells: empty, full or mixing cells. The empty cells do not contain obstacles inside. The full ones are completely filled by obstacles. The mixing contains some part filled by obstacles and some empty part. Mixing cells are gotten dividing the main frame by backtracking in cells until it gets a minimum size of cell [1] or until it gets either an empty or full cell. If we choose a good minimum size, enough to avoid obstacle, we have a reasonable processing time.

Starting on the principle that in the end of the process, the cells that had been divided are empty or full, a graph is created connecting the empty neighboring cells. To a cell be neighbor of another one a common point is enough. Later, it is executed a shortest path algorithm that uses Dynamic Programming Dijkstra (Cormen et al., 2001). In our graph, this algorithm gives the shortest path between two nodes (empty cells), giving an optimized planned trajectory for each robot, without collision.

The trajectory planned is converted to PWM levels and sent to the robots.

4 The Communication System

The CS broadcasts a set of packets containing the PWM levels and specific ID robot number. The robot owner of the packet must then extract the

---

[1] In this case, a minimum size mixing cell is gotten in a full cell.

PWM levels from the protocol and validate it, sending this information to the Control.

The transmission protocol consist of a header containing the owner of the packet and the PWM levels data. The information about the owner of the packet is sent n times by the workstation, so if a robot does not receive this information $\lambda * n$ [2] times, it is discarded. This approach is an attempt to treat noises on the wireless link.

After validation, the Communication Module signals the Control System on the arrival of a new PWM levels. Each robot has its own Communication Module.

5 The Embedded Reactive Control

The Embedded Reactive Control System is responsible for the reactive behavior, receiving low level sensor signals and sending the control to the motors and actuators. This system is composed by the main processor, power stage, motors, gearbox reduction, low level sensors and kicker signals.

The control receives the PWM levels data coming from the Communication Stage, process them and set the PWM signals to the motors. These signals are calculated based on a pre-calculated table with the voltage curve of each motor attached to its gearbox reduction.

Each robot has a kicker device. We use a very simple laser device to detect the ball. This reactive stage activate this device when some detection happens, enabling it only when the robot has the ball.

6 Implementation and Results

Building a robot team to play soccer is a big challenge in different fields. The range of technologies spans AI, robotic research and embedded system design. Therefore robots are ideal demonstrators for a number of research activities since they offer opportunities to evaluate various strategy theories, software algorithms, hardware-architectures and design techniques.

We have implemented our proposal with a very limited budget. The Furgbol system was developed in a computer with an Athlon 64 X2 4800+ processor and 2GB of RAM. The Furgbol software has been developed using GNU/Linux operational system and C++ programming language with the QtDesigner development tool [3].

The Deliberative Stage The workstation (CS) is connected to two digital cameras from Samsung (SC-D364 model) with IEEE1394 video outputs. Currently the cameras are connected on VIA1394 Firewire card, VT6306L chipset, with 6x4 input/output, operating with a transfer rate of 400 Mbps (50MB/s).

We are working with three new libraries: libraw1394 and libiec61883 that establishes the communication with the 1394 bus and carries through the data transference; and libdv, that allows the refinement of the received information. All steps of the Deliberative Stage were implemented. The Figure 4 shows a frame with the classified colors, segmented and localized agents and identified free cells. Notice that the field had been decomposed into full and empty cells. The vision system update each robot position in a 50ms rate sample.

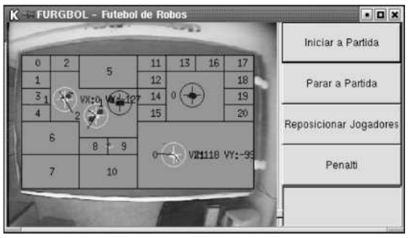

Figure 5: Interface

The Communication Stage The wireless communication is implemented with the Radiometrix's BIM2-433-64-S module, on the 433MHz frequency range. The workstation broadcasts the packets information about the PWM levels, with a bandwidth of 19200 bps. For instance, the CS sends two times the information about the owner of the packet. Each robot has also its own Communication Module, composed by the BIM2 Transceiver. Currently the communication is one-way only.

The Embedded Reactive Control The onboard processing is made by a low cost 16 bits RISC microcontroller from the DSPIC30F family by Microchip, running at 40MHz. The DSPIC family of microcontrollers has a wide range of applications to assist on the programming process. In our project the C programming language was chosen, using the Microchip's MPLAB environment to generate the assembly code.

The board is divided in three distinct stages: Communication Stage (detailed earlier in this section), Power Stage and Control Stage. The power circuitry consists of LMD18200T H-bridges by National.

Power is supplied by twelve AA NiMH batteries, each one able to deliver 1,2V/2800mAh. The Control Stage is responsible for the Actuation Step, which are implemented in the microcontroller program. Nowadays, we use three omni directional wheels in a 120 degree disposition. Let $F0$, $F1$ and $F2$ be the force vectors from each wheel, b the distance from each wheel to the mass center of the robot's chassis, w the robot's angular velocity, v the robot's velocity vector, r the wheel radius, and $w0$, $w1$, $w2$ the angular velocity of each wheel. The angular velocities are

---

[2] Being $0 \leq \lambda \leq 1$.

[3] Source codes avaible in www.ee.furg.br/FURGBOL

defined by equation $w_i = (v \cdot F_i + b \times w)/r$ for $i = 0, 1, 2$ (Reshko et al., 2000). These values are converted in PWM signals. A onboard pre-defined table maps from PWM setpoint to motor voltage.

We have designed and built the chassis, wheels and kicker device, see figure 5. The reduction gearboxes are able to rotate the wheels at 300 RPM with DC 12V motors, using omni-directional wheels. These wheels have a diameter of 50mm that makes possible to develop a maximum linear speed of 0.7m/s. In order to support the modifications over the old model, like new gearboxes, motors and wheels, a new chassis made out of aluminum was constructed 6. The total weight is 1.2Kg.

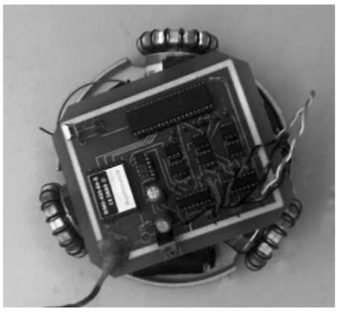

Figure 6: New FURGBOL robot structure.

All stages and algoritms run online (except the automatic calibration). See www.ee.furg.br/FURGBOL for a set of videos of FURGBOL performance.

7 Conclusions

RoboCup contest is an important test-bed for several areas of the Robotic and Computer Science and Engineering. In addition, for students it is a practical opportunity to develop knowledge in so many areas.

In this paper, we have described a low cost model underlying the FURGBOL Brazilian autonomous robot F-180 team, its implementation and our experiences with it. From a set of theories and algorithms, we have designed and implemented a real team of robots. We have proposed an architecture composed by three main modules proposed: *i.* a Deliberative Stage, *ii.* a Communication Stage and, *iii.* a Embedded Reactive Control. Relevant aspects of our architecture, like software, complexity, hardware and design issues are presented, detailed and analyzed. Our architecture was implemented using inexpensive and easily extendible hardware components and a standard software environment. And, even a very limited budge , FURGBOL has show to be a relatively successful approach; we are three times champion of Brazilian Robocup and vice-champion of Latin American Robocup twice.

We have a set of future short term and long term perspectives. For this year, we intend to improve our kick device, besides to test the robot's new chassis and wheels under real conditions. All of the recently-developed components have the potential to enhance our gameplay. We will continue to work intensively during the next two years, aiming to add local vision and embbebed treatment to our robots.

References

Amato, N. (2004). Randomized motion planning, *Technical report*, University of Padova.

Bailey, D. (2002). *A new approach to lens distortion correction, Proceedings Image and Vision Computing*, Proceedings Image and Vision Computing New Zealand.

Botelho, S. S. C. and Alami, R. (2000). M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement, *ICRA2000*.

Bruce, J., Balch, T. and Veloso, M. (2000). Fast and inexpensive color image segmentation for interactive robots, *IEEE IROS'2000*.

Cormen, T., Leiserson, C., Rivest, R. and Stein, C. (2001). *Introduction to Algorithms*, McGraw Hill/MIT Press.

D'Andrea, R. (2003). Robocup2003 cornell robocup documentation, mechanical group final documentation, *RoboCup 2003*, Springer.

Egorova, A., Gloye, A., Liers, A., Rojas, R., Schreiber, M., Simon, M. and Tenchio, O. Wiesel, F. (2003). Fufighters 2003 (global vision), *RoboCup 2003*.

Gonzalez, R. and Woods, R. (2001). *Image Processing*, Addison Wesley Pub.

Kuth, A., Bredenfeld, A., Günther, H., Kobialka, H., Klaassen, B., Licht, U., Paap, K. L., Plöger, P.G. Streich, H., Vollmer, J., Wilberg, J., Worst, R. and Christaller, T. (1999). Team description of the gmd robocup-team, *RoboCup-98: Robot Soccer World Cup II*.

Latombe, J. (1991). *Robot Motion Planning*, Kluwer Academic Publishers.

Loomis, J., Palmer, J. and Pandit, P. (2003). Performance development of a real-time vision system, *RoboCup 2003*, Springer.

Parker, L., Schneider, F. and Schultz, A. (2005). *Multi-Robot Systems. From Swarms to Intelligent Automata*, Springer.

Reshko, G., Nourbakhsh, I., Mason, M. and Zeglin, G. (2000). Holonomic motion control, *http://www.cs.cmu.edu/ pprk/physics.html*.

Shimizu, S. and Nagahashi, T. Fujiyoshi, H. (2005). Robust and accurate detection of object orientation and id without color segmentation, *RoboCup 2005*.