FURGBOL Team Description Paper

Silvia Botelho, Emanuel Estrada, Gisele Simas, Lorenzo Taddei, Marcelo Galeriano, Mariane Medeiros, Paulo Drews Jr, Rafael Deon, Renan Almeida, and Renato Neves

Fundação Universidade Federal do Rio Grande - FURG
Av. Itália Km 8, Rio Grande, RS, Brazil,
{galeriano, maninho, gisele, lorenzo, mariane, paulo, rafaeldeon, renan, renato, silviacb}@ee.furg.br
http://www.ee.furg.br/furgbol/

Abstract. The present paper describes a very low cost model underlying the FURGBOL Brazilian autonomous robot F-180 team, its implementation and, our experiences with it. The FURGBOL RoboCup team uses inexpensive and easily extendible hardware components and a standard software environment.
We propose an architecture which is composed by three main stages: i.A Deliberative Stage, ii A Communication Stage and, iii. A Embedded Reactive Control. This paper describes the relevant aspects of our architecture, like software, hardware and design issues.

1 Introduction

The field of multi-robot systems has become enlarged [12]. The robocup competition is an excellent test-bed for research in several areas related to the Computer Engineering and Science.

Several important approaches propose to build sophisticated multi-robot teams through the combination of expensive and complex hardware and mechanical devices [14, 7, 6, 11]. From an educational perspective, the RoboCup Competitions is also a great motivation for exposing students to design, build, manage, and maintain complex systems. However, nowadays, how to participate of a RoboCup Competition with a very limited budget?

The FURGBOL F-180 Team is an effort of the Department of Computer Engineering of the Fundação Universidade Federal do Rio Grande, Brazil, in order to foster research in robotics and artificial intelligence. FURGBOL team is composed by a group of undergraduated and graduated students. Our team use inexpensive and easily extendible hardware components and a standard software environment. For instance, so many components of our system are just parts removed from abandoned electronic equipment, as the motors for example, which are taken from CD-ROM drives. Besides, the FURGBOL platform is entirely based on open source software. Even 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 compose 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.

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 [3]. 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 1.

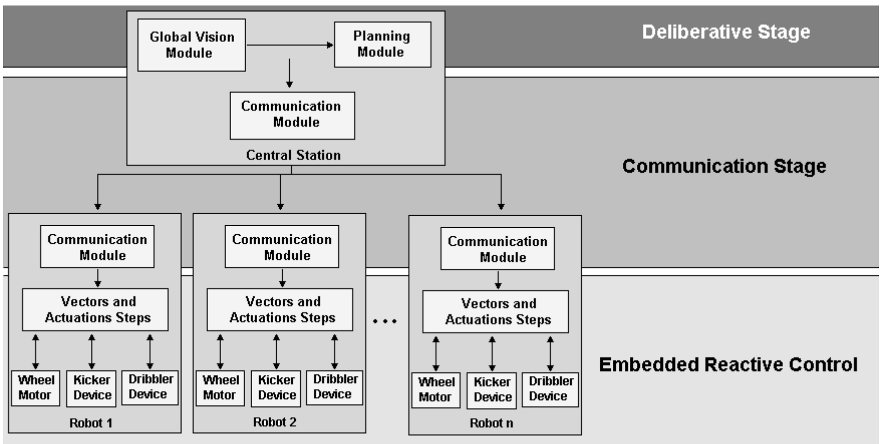

Fig. 1. Our architecture with three main Stages.

A Centralized Deliberative System is in charge of the global perception of the field and teams, identifying soccers and 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 video cameras located above the field. It 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 locate 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 [2]. The two following equations 1 are used to correct radial distortion.

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

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.

The HSV Color Transformations and Color Calibration Initially, the captured images are in RGB format (Red Green Blue). Aiming a more robust system in relation of luminosity variation, a HSV (Hue Saturation Value) color space transformation is performed [9]. In Robocup Rules, only a limited number of colors 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 calibration step where the H, S and V intervals are defined for each class. After this, the whole image is classified, pixel by pixel, through the H component. But, like some intervals of H are the same for different color classes, it is necessary to make a distinction among the classes by other pixel components, either S (saturation) or V (value), as described at [4].

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 pre-defined minimum size. At first, the position of all blobs found at one image are stored. So, it is started a process of conection 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 located inside of the boundaries of robots are conected to them and the remaining blobs are discarded. Finally, the orange blobs undergoes an anlaysis that determines the position of the ball into the field. Notice that there are two potencially problems in searching for orange blobs: *i.* even 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 at a determined radius around the blue and yellow blobs are discarded. In this case the last position of the ball from previous frame will be maintained [11].

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. 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.

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.

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 dribbler (see Figure 2 and 3).

Figure 2(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.

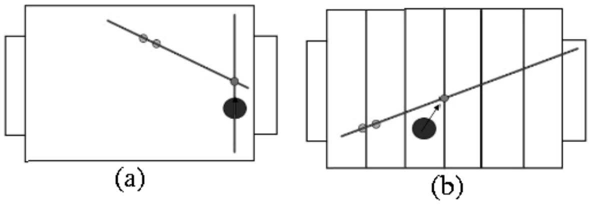

Fig. 2. (a) Goal-Keeper Strategy (b) Defense Strategy

The defense role uses as information the topological label of the ball, to move the robot to defense the goal, see Figure 2(a). Each area has a vertical straight line, wich represents its center. In Figure 2(b) we can see a red 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.

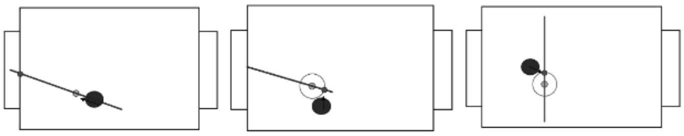

Fig. 3. First, second and third attack strategies, respectively

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 approach allows a planning of the trajectory of the robot in warranty that it does 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) [1].

The method of approximated cell decomposition as shown by [10] 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 of 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, is executed a shortest path algorithm that uses Dynamic Programming Dijkstra [5]. 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 Velocity Vectors and send to the robots.

4 The Communication System

The CS broadcasts a set of packets containing the velocity vectors and specific ID robot number. The robot owner of the packet must then extract the velocity vector from the protocol and validate it, sending this information to the Control.

The transmission protocol consist a header containing the owner of the packet and the data about the velocity vector and angular velocity. The information

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

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 ignore the interferences on the wireless link.

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

5 The Embedded Reactive Control

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

The control receives the velocity vector and angular velocity data coming from the Communication Stage, process it and send the PWM signals to the motors. The Control Stage is responsible for the Vectors and the Actuation Steps, which are implemented in the microcontroller program.

Vectors Step Converts the velocity vector and angular velocity sent by the workstation to the angular velocity of each wheel. This converting process relies on the mechanical configuration of the robot. The rotation velocities must then be converted to PWM signals for each motor, on the Actuation Module.

Actuation Step This step converts the angular velocities calculated by the Vectors Module in PWM signals. This conversion follows a pre-calculated table with the voltage curve of each motor attached to its gearbox reduction.

The Low Level Sensor Each robot has a kicker and dribbler device. We use a very simple laser device to detect the ball. This reactive stage actives these devices when some detection happens, enabling then only when the robot has the ball.

6 Implementation and Results

We have implemented our proposal with a very limited budget, about US$300,00 per robot. Basically, the Furgbol system was developed in a computer with an Athlon XP 2400 processor and 512MB of RAM. The Furgbol software has been developed using GNU/Linux operational system and C++ programming language with the QtDesigner development tool.

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).

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

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 allow 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. It can be observed that the field had been decomposed into full and empty cells.

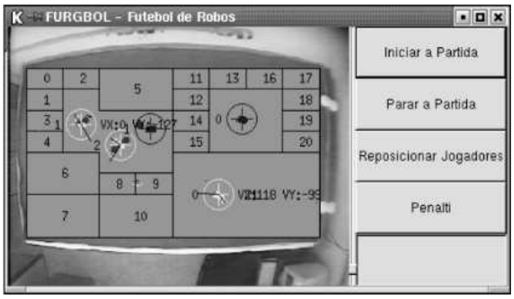

Fig. 4. Interface

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 velocity vectors and angular velocity, with a bandwidth of 19200 bps. For instance, the CS sends six time 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 8 bits RISC microcontroller from the PIC16F877 family, running at 20MHz. The PIC 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 CC5X compiler from B Knudsen Data, Norway. It integrates the Microchip's MPLAB environment. A research for the development of a new onboard system based on the DSP (Digital Signal Processor) platform is on course.

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 L293D H-bridges isolated from the remaining circuitry by 4N25 optocouplers. Power is supplied by eight AA NiMH batteries, each one able to deliver 1,2V/2800mAh. The Control Stage is responsible for the Vectors and the Actuation Steps, 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 defined by equation $w_i = (v \cdot F_i + b \times w)$ for $r, i = 0, 1, 2$ [13]

Vectors Step is implemented onboard, calculating these equations. We have also implemented the **Actuation module**, which has a pre-defined table with the voltage curve of each motor attached to its gearbox reduction.

Mechanical design The FURGBOL Robot, see Figure 5 uses simple equipment as motors, reduction gearboxes and chassis available in the market or adapted from other equipment.

Fig. 5. Current FURGBOL robot design.

The reduction gearboxes are able to rotate the wheels at 160 RPM with DC 9V motors, using omni-directional wheels by North American Roller Products (NARP). These wheels has an diameter of 40mm and plastic rolling (Carter and et al, n.d.) that makes possible to develop a maximum linear speed of 0,36m/s.

A mechanic kicker device has been developed, making use of the elastic potential energy of a spring and a linear actuator for its compression. This option was adopted for its easier development and low cost. Another advantage of the mechanical kick device is the low electromagnetic interference generated, in comparsion to the solenoid solution.

Also, a dribbler device has been developed to keep the ball in kick position in the front of the robot [6]. The prototype has been developed using old printer parts.

The current chassis design is composed by a thin aluminum base with a plastic coverage. This structure has showed to be good enough by being very light. However, with kicker and dribbler devices integrated at chassis structure, the current structure becomes inadequate. Then it has been developed a new structure and a new set of wheels with aluminum and TecNil rests, as shown in Figure 6.

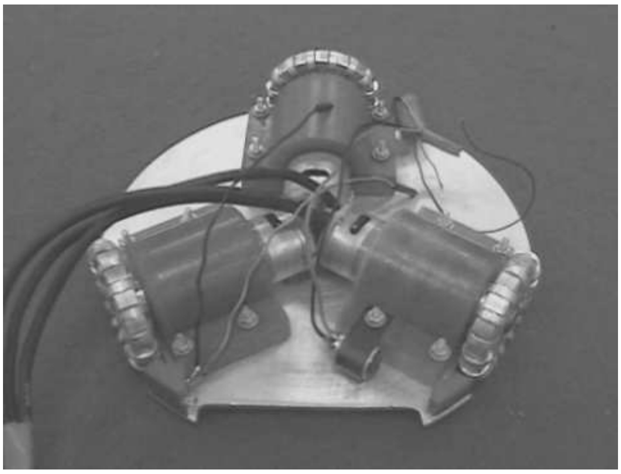

Fig. 6. New FURGBOL robot structure.

7 Conclusions

RoboCup contest is a 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 ways. So, in this paper, we have described a very low cost model underlying the FURGBOL Brazilian autonomous robot F-180 team, its implementation and our experiences with it. We have propose an architecture composed by three main modules: *i.* a Deliberative Stage, *ii.* a Communication Stage and, *iii.* a Embedded Reactive Control. Relevant aspects of our architecture, like software, 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 (U$ 1500,00), 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 the year of 2007, we intend to improve the vision system with clustering algorithms for color calibration and **RLE** (Run Length Encoding) compress image algorithms for segmentation [8]; what concerns to hardware, we inted to implement a two-way communication, change PIC microcontroller to faster DSP, DSPic for example, as well use a dedicated PIC microcontroller for communication. Besides, we intend to use the new prototype in development, with kicker, dribbler and ball detection devices connected, either.

References

1. N. Amato. Randomized motion planning. Technical report, University of Padova, 2004.
2. D. Bailey. *A new approach to lens distortion correction, Proceedings Image and Vision Computing*. Proceedings Image and Vision Computing New Zealand, 2002.
3. S. S. C. Botelho and R. Alami. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In *ICRA2000*, 2000.
4. J. Bruce, T. Balch, and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. In *IEEE IROS'2000*, 2000.
5. T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw Hill/MIT Press, 2001.
6. R. D'Andrea. Robocup2003 cornell robocup documentation, mechanical group final documentation. In *RoboCup 2003*. Springer, 2003.
7. A. Egorova, A. Gloye, A. Liers, R. Rojas, M. Schreiber, M. Simon, and F. Tenchio, O. Wiesel. Fu-fighters 2003 (global vision). In *RoboCup 2003*, 2003.
8. L. A. Gomez. Sistema de vision para el equipo de robots autonomos del itam. Master's thesis, Instituto Tecnologico Autonomo de Mexico ITAM, 2004.
9. R. Gonzalez and R. Woods. *Image Processing*. Addison Wesley Pub., 2001.
10. J. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
11. J. Loomis, J. Palmer, and P. Pandit. Performance development of a real-time vision system. In *RoboCup 2003*. Springer, 2003.
12. L. Parker, F. Schneider, and A. Schultz. *Multi-Robot Systems. From Swarms to Intelligent Automata*. Springer, 2005.
13. G. Reshko, I. Nourbakhsh, M. Mason, and G. Zeglin. Holonomic motion control. In *http://www.cs.cmu.edu/ pprk/physics.html*, 2000.
14. S. Shimizu and H. Nagahashi, T. Fujiyoshi. Robust and accurate detection of object orientation and id without color segmentation. In *RoboCup 2005*, 2005.