FURGBOL TEAM DESCRIPTION PAPER

André Núñez*, Eduarda Almeida*, Emanuel Estrada*, Gisele Simas*, Guilherme Fickel*, João Pedro Longhi*, Leandro Morita*, Luan Silveira*, Lucas Novelo*, Thiago Santos*, Silvia Botelho*

**FURG - Universidade Federal do Rio Grande*

Emails: afnunez@gmail.com, dudahalmeida@hotmail.com, emanuelestrada@gmail.com, gisele_simas@yahoo.com.br, guilhermefickel@gmail.com, joaolonghi@gmail.com, guerilla_roots@hotmail.com, luansilveira.dp@gmail.com, lucasnovelo@gmail.com, tssantos@gmail.com, silviacb@furg.br

Abstract— The present paper describes the FURGBOL robot F-180 team. The FURGBOL RoboCup team uses inexpensive and easily extendible hardware components and a standard linux software environment. We propose a global vision system and a modular architecture, having three main stages: *i.* a Deliberative Stage (associated with strategy and path planning issues), *ii.* a Communication Stage and, *iii.* an Embedded Reactive Control. We describes the relevant aspects of our architecture like software, hardware and design issues.

Keywords— robotics, computer vision, multiagent cooperation.

Resumo— Este artigo descreve a equipe de futebol de robôs FURGBOL, participante da categoria F-180. A equipe FURGBOL utiliza componentes de hardware de baixo custo, facilmente extensível e um ambiente de software sob a plataforma Linux. Propomos um sistema de visão global e uma arquitetura modular, tendo três fases principais: *i.* estágio Deliberativo (associado a questões estratégia e planejamento de trajetória), *ii.* estágio de Comunicação e *iii.* Controle Reativo Embarcado. Neste artigo, descrevemos os aspéctos relevantes da nossa arquitetura como software, hardware e outras questões de projeto.

Palavras-chave— robótica, visão computacional, cooperação multiagente.

1 Introduction

The field of multi-robot systems has become enlarged (Parker et al., 2005). RoboCup is a long-term effort of the academic and industrial research community to develop teams of robotic football/soccer players. Issues associated with accurated motion, team coordination, computer vision, communication, embbeded systems must be treated, regarding real time restrictions. (Shimizu and Nagahashi, 2005; Egorova et al., 2003; D'Andrea, 2003; Loomis et al., 2003).

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 a great motivation for exposing students to design, build, manage, and maintain complex robotic systems.

The FURGBOL F-180 Team is an effort of the Center for Computational Science of the Universidade Federal do Rio Grande, Brazil. Our goal is to stimulate research, teaching, and applications in the fields of artificial intelligence and collaborative robotics. Our team use inexpensive and easily extendible hardware components and a standard linux software environment. Besides, the FURGBOL platform is entirely based on open source software. Even a very limited budget, FURGBOL has show to be a relatively successful approach; since it started, in 2001, we are five times champion of Brazilian Robocup, vice-champion of Latin American Robocup twice, and last year FURGBOL wins the Latin American Robocup.

This paper describes a set of 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 omnidirectional 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.

The chassis consists of one laser-cut aluminium plate, having a diameter of 176mm and a height of 145mm, see figure 1.

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 reac-

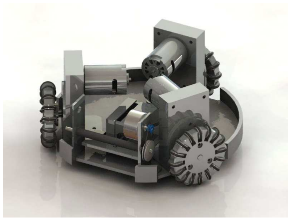

Figure 1: The CAD chassis of FURGBOL robot.

tive control and a central coordinated decision for incremental plan.

A Centralized Deliberative System is in charge of the global perception of the field and teams, path planning and the robot behavior. The communication system exchanges informations 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

It is assumed 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. This stage has two main modules: the Global Vision Module and the Planning Module.

3.1 The Global Vision Module

In order to provide the information not only of the robots but of all the game scenario, the global vision was adopted. Global vision works with a set of n cameras fixed above the field of play, giving the pose of robots, ball and all the relevant visual information. Through a set of image processing techniques, the system performs a tree main steps: *i.* segmentation; *ii.* color classification based either on the HSV or normalized RGB color space analysis; and *iii.* agents localization. These steps are described as follows next.

The Color Classification

The images acquired from the cameras are captured in RGB (Red Green Blue) format. Our proposal uses a normalized RGB approach, in wich one of the normalized components assumes the minimum value (0), a second component assumes the maximum value (1) and the third one assumes an intermediate value between 0 and 1.

In this context, a color circle divided into intervals with well defined boundaries is used (Oliveira, 2007). Considering the minimum and maximum normalized components, the chosen interval defines two possible colors as candidates to the correct classification. At this point, a second intermediate component is located inside this interval and thus the distance of this component is measured in relation to the interval boundaries. The minor distance indicates the closest boundary, selecting which of two colors is definitely the right one.

Image Segmentation This stage is actually started before the color classification. Based on previous acquired frames from the empty field of play, each new frame is subtracted of the empty ones (Cerqueira et al., 2006). This technique reveals the areas of the new image substantially different of the old ones in view of diminish the amount information to be processed. After the image subtraction, only the pixels considered relevant to the process undergoes color classification. These pixels are grouped into previous defined intervals of colors named color classes. These classes are of two types: (i) primary classes: orange (ball color), blue and yellow (team colors); (ii) secondary classes: light green, cyan and light pink. With the classification finished, the image scan is started and each pixel belonging to one of the described color classes starts a recursive process in search of similar pixels. This process is known as region growth, and near pixels belonging at the same color class defines a *blob*. As long as all blobs are found, the image segmentation is done.

Agents Localization Finally, all blobs found in the segmentation stage have their center calculated in an effort of perform the blobs bellonging to the process. This process connects some secondary blobs to one primary blob through the search of blob centers whose distance to the center of a main blob is smaller than the radius of the robots. In this way, all robots are recognized and have their positions defined, as well the ball. All the information extracted throughout the vision system is sent to the strategy system which will plan and take decisions that will guide robots through the field of play.

There are problems with the correct position estimation. In some cases, on vision, the ball is in front of the robot, however the ball isnt́ there really. The solution for this problem is simple. Considering the robot and camera heights, we calculate, with pythagorean theorem, the real distance between the robot and the ball.

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, like in (Laue et al., 2007). 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) are defined based on the relative positions between robots and ball.

In addition, 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. Three kinds of roles are defined: the goal-keeper, the defenser and the attacker. Each role has a own state machine.

Each one of these three basic roles supplies a target position to where each robot must move itself. In addition, the planning module decides when a robot should rotate or activate the kicker and dribbler devices. These actions happen when the robot is close enough of the ball to lead or kick it. In order to dribble or to kick, the robot must turn to position the devices in front of the ball. The robot spins if the angle between its front and the correct position is less than a constant.

A Path Planning Step The Path Planning Step defines the robot motion to arrive to target position avoiding obstacles.

Thus, it is first checked whether a straight-line trajectory to the target position is possible without any collision. Then the trajectory and obstacles in the field are regarded as polygons. If one of the vertices of any polygon that represents an obstacle is inside the trajectory polygon, there will be a collision.

If the straight-line path is not possible, we apply the approximated cell decomposition method. This approach allows a robot trajectory planning without any collision.

The method 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 (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.

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

4 The Communication System

The CS broadcasts a set of packets containing the PWM levels, dribbler and kick informations and specific ID robot number. The robot owner of the packet must then extract the PWM levels from the protocol and validate it, sending this information to the Control.

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.

6 Implementation and Results

We have implemented our proposal with a very limited budget. The Furgbol system was developed in a computer with an Intel Core 2 Quad 2.4GHz 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 [2].

The Deliberative Stage The workstation (CS) is connected to two digital cameras from AVT GUPPY(model:F033C) with IEEE1394 video outputs. Currently the cameras are connected on VIA1394 Firewire card, VT6306L chipset, with 6x6 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. The vision system update each robot position in a 20ms rate sample.

[2] Source codes available in www.ee.furg.br/∼furgbol

The Communication Stage The wireless communication is implemented with the Radiometrix's BIM2-433-64-S module and BIM3A-914-64 module, at 433MHz and 914MHz frequency range respectivelly. 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, running at 30MHz. 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.

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 defined by equation $w_i = (v \cdot F_i + b \times w)$ for $r, i = 0, 1, 2$ (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 design and build chassis, wheels, dribbler and kick mechanic devices, see figure 2. 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. This weight is 1.2Kg. All stages and algorithms run online (except the automatic calibration). See www.ee.furg.br/∼furgbol for a set of videos of FURGBOL performance.

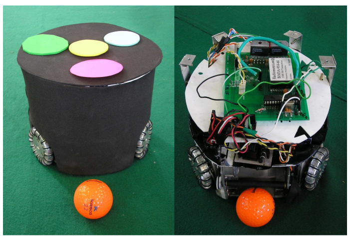

Figure 2: New FURGBOL robot structure.

7 Conclusions

RoboCup contest is an important test-bed for several areas of the Robotic and Computer Science/Engineering. 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. 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 five times champion of Brazilian Robocup and champion of Latin American Robocup last year, and vice-champion of this same tournament.

References

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

Cerqueira, A. C. T., Lins, F. C. A., Medeiros, A. A. D. and Alsina, P. J. (2006). A versão 2006 da equipe poti de futebol de robôs, *Jornada de Robótica Inteligente*.

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). Fu-fighters 2003 (global vision), *RoboCup 2003*.

Laue, T., Burchardt, A., Cierpka, K., Fritsch, S., Göde, N., Huhn, K., N., Kirilov, T., Lassen, B., Miezal, M., Lyatif, E., Schwarting, M., Seekircher, A. and Stein, R. (2007). B-smart team description for robocup 2007, *RoboCup 2007*.

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

Oliveira, L. M. (2007). *Segmentação fuzzy de imagens e vídeos*, Master's thesis, Universidade Federal do Rio Grande do Norte.

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*.