UChile1 Team Description Paper

Javier Ruiz-del-Solar, Juan Cristóbal Zagal, Pablo Guerrero,
Paul Vallejos, Christian Middleton and Ximena Olivares

Department of Electrical Engineering, Universidad de Chile
http://www.robocup.cl
{jruizd,jzagal,pguerrer,pvallej}@ing.uchile.cl

Abstract. The U-Chile-1 four legged team is an effort of the Department of Electrical Engineering of the University of Chile in order to foster research in robotics at high level. U-Chile-1 was composed this year by a group of undergraduate and graduate students of electrical engineering and computer science of our engineering faculty. This document describes the relevant aspects of the software which was developed from scratch by our team. The system has shown to be a relatively successful approach. For example we tied the game against Team Sweden and we won against the Turquey – Bulgarian Teams during the RoboCup 2003 competitions. We also had a satisfactory participation during the RoboCup Challenges. We consider this as a good result considering the short period of preparation.

1 Introduction

As a side effect of research activities, robotics is emerging as a new way of teaching sciences and technology; it allows students to apply a great amount of knowledge to a variety of problems which arise when solving robotics problems. At our department of electrical engineering a main interest is to foster the research of new wave robotics. Mobile robotics research is intended to be the main subject of research within our robotics laboratory.

RoboCup is a great opportunity to share new knowledge in the field as well as to compare the level of the universities by means of their performance during the games.

We believe that it is possible to have a good participation in this kind of tournaments even with our very restricted budged, compared to other universities within the league.

Research Interest of the Group

The interest of our group is on mobile robotics, autonomous systems, intelligent systems, and computer vision. We are currently focusing our efforts towards evolutionary robotics and collaborative robotics. We intend to contribute to these research areas by experimenting with groups of intelligent robotics agents, which are able to interact socially, to learn from their own experience and from their direct contact with the real world. Our long-term goal is to develop new approaches for robot machine learning having groups of mobile robots as our main test board. The RoboCup challenge fits perfectly in this context. It allows us to gain and share a great amount of experience in the difficult task of having a group of autonomous robots collaborating in a soccer team under standard game conditions. We consider that in order to improve further the skills of any of such group of robots, it is necessary to incrementally incorporate learning mechanisms aiding the engineering task. We aim at contributing with this great challenge by proposing novel learning strategies on each one of the main issues of the RoboCup four-legged soccer robot problem. As all of us know, the four-legged soccer is a very constrained problem in terms of hardware (camera, processor, available memory, sensors, etc.) and game field, in which many different processing approaches have been already proposed. As a new team in this league we want to specialize us in the introduction of evolutionary, adaptive and collaborative methods for the perception, control and strategy subsystems. For example, in the case of the visual object recognition problem, we have proposed a genetic-based system for the selection and tuning of rules for the detection of the ball, landmarks and goals [6]. We are now using this system for deriving rules for the detection and pose estimation of the robots in the game field. We are also exploring how learning can be applied for improving the strategy of our team. We believe that it is possible to learn from the real interaction between robots, minimizing the simulation on the learning process. Another topic in which we are working on is the adaptive learning of new strategies of localization by minimizing the observational cost (movement path of the robot head). We have one PhD student writing his thesis on evolutionary robotics, and one master student writing her thesis on evolutionary-collaborative robotics. They are using legged robots and the soccer problem for demonstrating and testing their proposed approaches. The main idea is to learn from real game experiences minimizing the simulation. In a near future we are going to explore the evolution of key movements of the robots (not gaits which have been broadly explored[3][4]).

Finally, we would like to express that we have a very good experience in the use of robot contests for fostering research activities on robotics and for motivating the interest of students in the field. In fact our research group was created as a consequence of the *1st IEEE Latin American Student Robotic Contest*, which we organized last year (http://www.ewh.ieee.org/reg/9/robotica/1stRobotContest). We are consolidating this group thanks to the creation of our two robot soccer teams, and our participation on RoboCup 2003. We believe it is important that a team of South America have the opportunity to participate in RoboCup 2003, for spreading robot soccer in a continent with a great potential in research and in which soccer is by far, the most popular sport. Through our work in the *IEEE Latin American Council* (http://ewh.ieee.org/reg/9/robotica/) we have seen an increasing interest of engineering students on robotics. We are organizing the *2nd IEEE Latin American Student Robotic Contest* in Brazil, which include some robot soccer contests (http://www.ewh.ieee.org/reg/9/robotica/2ndRobotContest).

2 System Architecture

Our system is divided into five task-oriented modules which are vision, localization, low-level strategy, high level strategy and motor action, see figure 1. The vision module operates on each robot locally. The localization module is distributed, it operates on each robot and a global estimate of the overall localization is generated in a distributed fashion. The high-level strategy module is also separated into a global part, which resides into the goalie player, and also a part which operates locally. The motor action module operates locally on each robot. Another component of our architecture is the wireless inter robot communication system. The following is a description of each one of these modules.

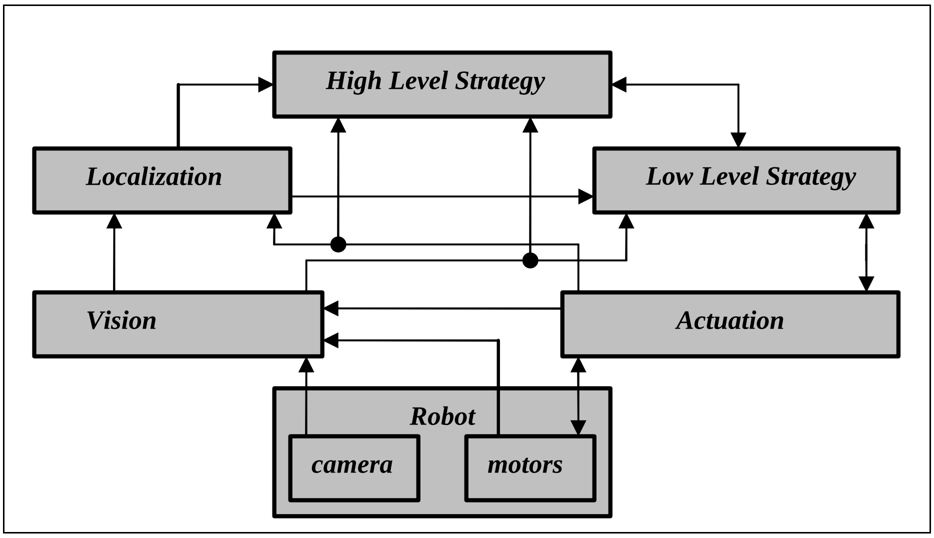

Fig. 1. Modular organization of our system. In the bottom the low level processes of vision, motor action and localization. On top the high level processes of strategy. We distinguish between local process (left) and global processes (right).

2.1 Vision

The vision module receives two low level inputs which are the raw image from the robot camera and state angles from the robot motor controllers of the head. Estimates of body inclination are also obtained from the actuation module. The output of this module, which is transmitted to the localization and low level strategy modules, consists on a list of relevant game field objects which detected into the image. For each one of these objects an estimate of their camera-relative and robot-relative coordinates are provided as well as estimates of their detection certainty. These relevant objects correspond to the red or blue robot players, any one of the six colored beacons, the cyan and yellow goals and the orange ball. For each detected object their distance and azimuth, projected over the field plane, are estimated, as well as corresponding confidence degrees for their detection, distance, and azimuth estimations. This module is divided into six processing sub-modules, which are: color segmentation, run-length encoding of the segmented image, labeling of connected regions, region extraction and characterization, rule-based object perception, and finally computation of distances, azimuth and projections of objects over the game field plane.

2.1.1 Color Segmentation

This sub-module takes as input, from the AIBO camera, the raw color image with their three channels Y, U, and V, and generates as output a single channel image containing identifiers for the background and the relevant colors of the game field, which are pink, yellow, green, cyan, blue, red, and orange. We have recently incorporated in this implementation the detection of white. The segmented image is then generated by extracting identifiers from a look-up table of 64 levels in each dimension Y, U and V. The values Y, U and V of each pixel are converted to indexes in the range 0 to 63 for accessing the look-up table values. The user-based color calibration process, which generates the look-up table, consists on first taking pictures from the game field and the environment with the AIBO camera. Then image regions, which are related to the colors of the game field, as well as examples of non relevant colors, which are used for training the background class, are selected within a supervised process. Training a special class for the non-relevant colors is a fundamental consideration in our implementation. Then, these sample pixels are extracted with the color identifier which is provided by the user. These samples are directly located into the look-up table according to their coordinates in the YUV space. This process gives rise to the generation of clusters into the YUV space. We have observed that the shape of these clusters is usually oblong, in general difficult to be characterized in terms of rectangles. Instead of doing so, we just leave these shapes as they are, but we enforce the shape and internal structure of these clusters by applying a median mask over the look-up table values and morphological operators of dilation and erosion over the clusters. Dilation serves for filling in the interior of clusters, while dilation serves for keeping them under their original size. The median mask is used for cleaning up the interfaces between clusters. In general this process requires few images of the game field, and it takes about 15 minutes.

2.1.2 Run-length encoding

This sub-module takes as input the resulting image of the color segmentation sub-module, and generates a run-length codification of the image in the form of an array of 4xN elements, where N is the variable number of segments which are extracted from each frame. A run or segment consists of a set of foreground connected pixels of the same color identifier within the same image row. This codification is generated by scanning the image thought its rows and, for each run or segment, the coordinates x and y of the first pixel are stored in the array as well as the corresponding length of the run. The UNSW team uses this codification [1], we consider that it is particularly useful for reducing the amount of data as well as for simplifying the subsequent labeling stage.

2.1.3 Labeling of Connected Regions

A variant of the connected components labeling algorithm is used for assigning a label to each connected segment, and for defining connected regions on the image. This process takes as input the 4xN run-length codification updating the fourth field of each segment with a corresponding label. This is performed by looking, on each row, the upper neighbors of each segment , and then copying their corresponding labels. If a segment has two upper neighbors having different labels, their corresponding labels are replaced going upwards along each branch. If a segment has no label, a new label is generated. In this process we consider only the foreground segments. Similar labeling processes have been proposed by other teams. We consider that this is a good choice in combination with the run-length codification.

2.1.4 Region Extraction and Characterization

The region extraction sub-module takes as input the run-length encoded and labeled image, and generates a list of regions or blobs which are related to each color. Each region is characterized with a set of region descriptors which are: the coordinates of the region bounding box, the coordinates of the region center of mass, the perimeter of the region, the mass of the region, the color of the region, and other descriptors. These descriptors are afterwards used for the recognition of relevant object into the game field.

2.1.5 Rule Based Object Perception

The task of the object perception sub-module is to identify image regions which are related to the relevant game field objects. The recognition of objects is performed by evaluating the response of a set of rules which are applied over a set of candidate regions or combinations of these candidate regions. For example the detection of a landmark is performed by evaluating the response of some rules to the combinations of regions of the colors pink, cyan, yellow, and green. The detection of the other players into the game field is also performed by applying rules over combination of regions. The detection of a ball usually requires that the related blob has the color of the ball, and if this is not the case one might expect to reject this candidate blob. The rules which are used in our team where derived by using our proposed genetic based method for visual rule selection and parameter tuning [6].

2.1.6 Computation of Distances, Azimuth and Projections

The distances and the azimuth of the objects are initially calculated with respect to the optical axis of the camera. These measures are then projected over the game field plane by simple matrix operations. These transformations are a function of the actual pose, described with three angles, of the robot head.

2.2 Localization

This module is in charge of estimating the localization of robots and ball into the game field. This information is used by the low and high level strategy modules and it can also be shared among the robots of the team.

The localization module corresponds to an implementation of the Monte Carlo Localization Method (MCL). We have selected this method since, among other methods, it shows to me more robust allowing for an efficient fusion of the data which comes from the vision and actuation modules. One drawback of our implementation is that it requires of a large convergence time. This result in that players of our team were quite affected each time they were manually transported from one place of the field to another when doing an obstruction, for example. After teletransportation our robots were quite confused on their localization.

Since this is a mayor problem in our implementation we are currently working on an implementation of the Adaptive-MCL method. Early results of this new implementation have confirmed our expectations, A-MCL converges quickly to good solutions when the robot is being kidnapped, however the steady state solutions of this method are still below the corresponding performance of the classical MCL [2].

From the beginning of our work we intended to share the information of the position of the ball among the players of our team. However we did not use this capability during RoboCup 2003 due to some implementation problems.

The estimates of the localization uncertainty which are obtained from the method are used for triggering behaviors of the robot head which help the robot to localize itself. For example the robot fixes its sight towards certain landmark in order to validate its current position.

2.3 Actuation

This module is in charge of controlling the robot motors according to instructions provided by the low-level strategy module and it is also in charge of generating odometric estimates of the robot displacements.

The module, according to figure 1, receives low level inputs from the robot motor controllers, and exchanges information with the localization and low-level strategy modules. The actuation module generates outputs for the motor controllers, an also an estimate of the robot inclination for the vision module. It provides odometric information to the localization and low-level strategy modules. It receives instructions from the low-level strategy module.

The motion of the head and body limbs are controlled independently, there are some instructions for the head, another set of instructions for the body limbs, and there are also some instructions for the overall robot. A head instruction consists on the specification of the three head angles. A body instruction consists on the specification of an infinitesimal change in the robot pose.

Each instruction is internally decomposed into a set of microinstructions. A microinstruction consists of a vector of length 15 in which the angular state of each robot joint is specified as a reference, it also contains odometric information and timestamps for ensuring the synchrony of motions. A microinstruction is divided along the time into 16 frames, each one having duration of 8 milliseconds. For each frame, a motor reference value is computed as a linear interpolation among the references of the previous and current microinstructions.

The decomposition of instructions into microinstructions is performed in two different ways; first if the instruction is a set of static matrices specifying a motion itinerary, the decomposition is performed by interpolating the different reference points on each joint. On the other hand, if the instruction is part of a parametric walking, then the microinstructions are computed internally depending on the desired displacement. It is important to notice that in our implementation the robot is free to move on an omni directional fashion, each joint variable is calculated in real time in order to account for an arbitrary body pose change, this is performed by calculating the inverse kinematics of each joint and the displacement of each limb extreme which is necessary for producing an arbitrary robot displacement.

2.4 Low Level and High Level Strategy

The strategy module is conceptually divided in two modules; the low level strategy module which is in charge of producing behaviors of the robot and the high level strategy module which is in charge of triggering these behaviors according to some higher level game circumstances.

2.4.1 Low Level Strategy; Behaviors

This module is always connected to the high level module and it exchanges information with the actuation module, it receives information from the vision module and from the localization module.

In this module a set of robot behaviors is defined. A behavior consists of a certain function of robot states and sensor states which produces instructions for the actuation module. These behaviors are executed instantaneously, and they do not consider long term decisions. A behavior should be called several times in order to achieve its corresponding goal. Our implementation is based on eight behaviors which are:

1. *Go to ball and kick it towards some target.*
2. *Go to a certain target with some orientation.*
3. *Place itself among two points with some orientation.*
4. *Do tracking of some object.*
5. *Help to localization.*
6. *Do an ellipse with the head.*
7. *Do nothing.*
8. *Move the head backwards.*

2.4.1 High Level Strategy

In this module different behaviors are triggered according to the strategy of each player and the state of the game and the color of the team. We distinguish four states for the players which are: Initial state, Playing, Penalization state, and final state. These states can be established with the game controller of by pressing some buttons on the robot. For example in the initial state localization behaviors are triggered. During the Playing state a decision tree is followed according to the player role. We define two different roles for the robots, which are Goalie and Attacker.

Goalie Role

The goalie tries to defend the goal and block any attacker which comes closer to the goal area. This is performed with the decision tree which is shown in figure 2.

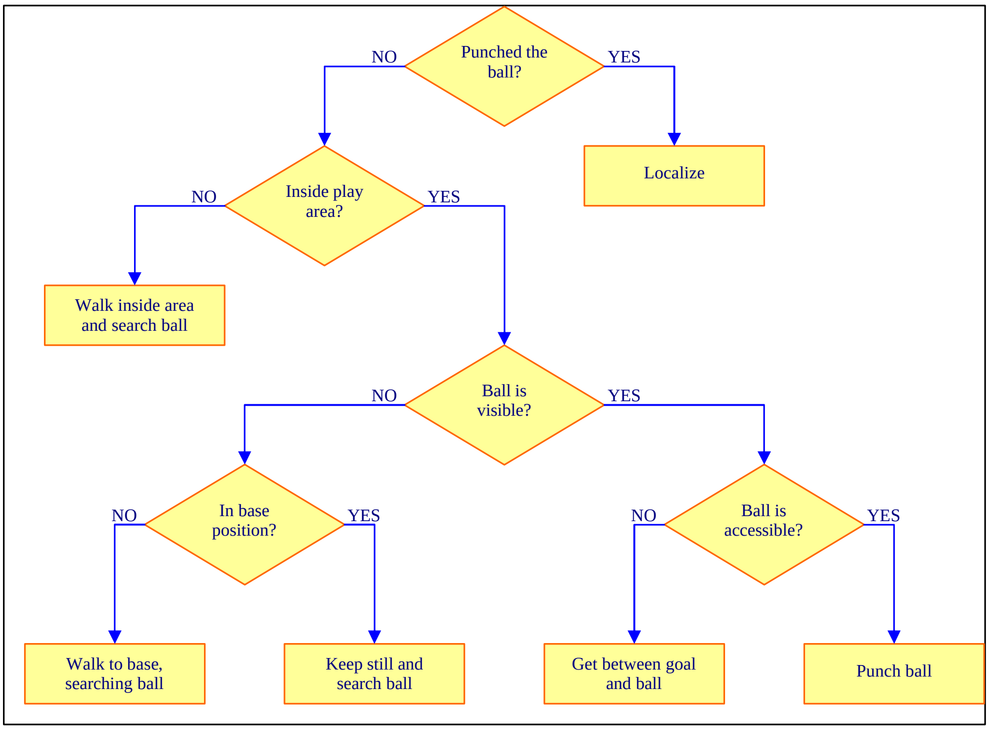

Fig. 2. Illustration of the goalie decision tree.

Attacker Role

The main objective of the attacker is to be able to find the ball, and then to punch it towards the goal of the opponent team. The first part of the attacker decision tree, which is shown in figure 3, is for determining the position of the robot with respect to the ball and to decide whether the robot is able to catch the ball.

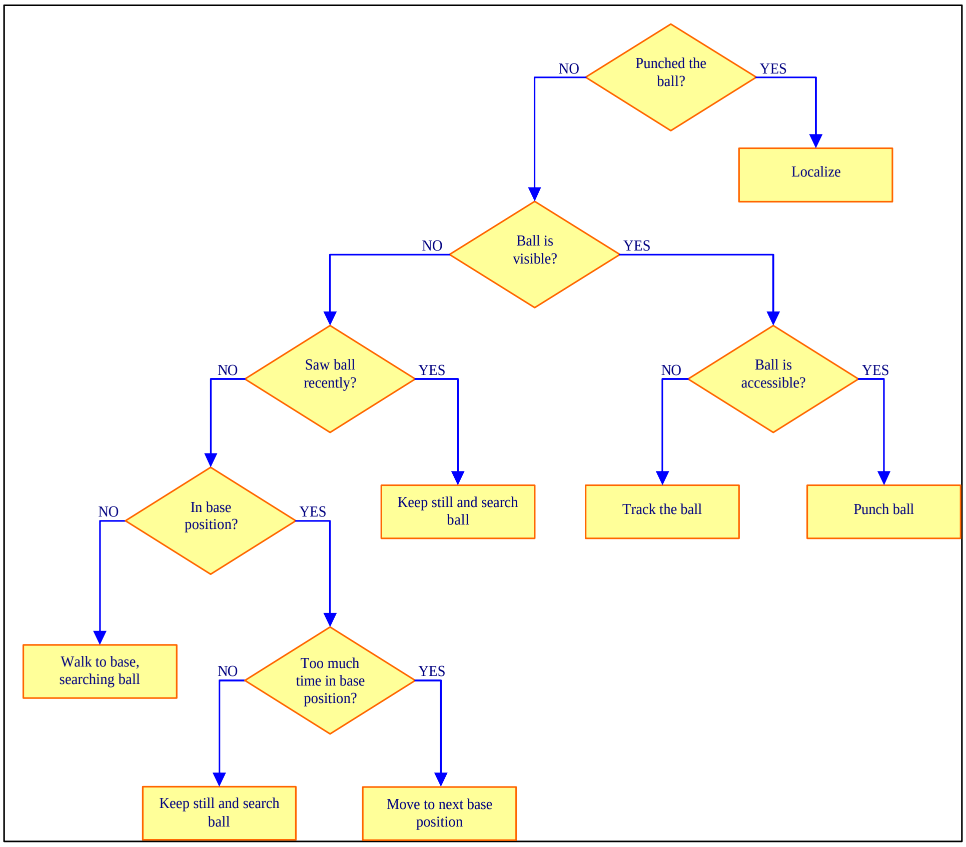

Fig. 3. Illustration of the attacker role.

Another difference between the *Goalie* role and the *Attacker* role is that the *Attacker* role defines two base positions for each player, instead of just one. This allows a player to be able to guard an area, looking for the ball nearby a position in the field, and if the robot has stayed for too long in one place, it must go to the other base position and look for the ball near the new position. These two base positions are inside the play area defined for the player.

2.5 Software development organization

We noticed, from the very beginning of our work, that developing software for the RoboCup Four Legged Challenge also involves establishing a fully organized and scalable working platform allowing the interaction of many people over the same piece of source code. The following are the main considerations that we have taken with respect to the software development organization of our team:

- We use the Concurrent Version System (CVS) to allow many people working over the same source code without having problems while merging versions, as well as having a good managing of the code versions.
- The code organization should exploit the advantages of the open-r object oriented approach; the code is divided into modular pieces under an architecture which allows scalability.
- We have dedicated a great amount of attention to the design of efficient interfaces between the modules.
- The code is written in a way that allows it to be compiled under both Aperios and a debugging platform such as Linux or Windows.
- The code is written in a way that facilitates the incorporation of monitoring tools.

3 Background of the principal investigator

The main background of Prof. Dr. Javier Ruiz-del-Solar is on image processing and computer vision. He has contributed to the fields of *Face Analysis*, *Application of Soft-Computing technologies to Computer Vision and Pattern Recognition problems*, *Texture Analysis*, *Retrieval* and *Synthesis*, and *Robotics and Autonomous Systems* with more than 60 articles published in international journals, conferences and books (see http://www.cec.uchile.cl/~jruizd/publications.html). Since 2002 he serves as technical reviewer of the *IEEE Trans. on Systems, Man and Cybernetics – Part B*, and in the past was reviewer of *Pattern Recognition Letters*.

Since 2002 Prof. Ruiz-del-Solar has been involved in the organization of robotics contests in Latin America and in the increasing of robotics activities in Chile. He is creator and chairman of the *IEEE Latin American Robotics Council* (http://ewh.ieee.org/reg/9/robotica/), creator and director of Electrotechnology Laboratory of the Universidad de Chile (http://etec.li2.uchile.cl/home.html), which includes a robotics laboratory, chairman of *1st IEEE Latin American Student Robotic Contest* (Nov. 29 – 30 2002, Santiago, Chile), chairman of the *2nd IEEE Chilean Student Robotic Contest*, (August 8 – 9, 2003, Santiago, Chile), and organizer of the *2nd IEEE Latin American Student Robotic Contest* (Sept. 13-14, 2003, Bauru, Sao Paulo, Brazil). Regarding the organization of other international scientific events, Prof. Ruiz-del-Solar was chairman of the *2nd International Conference on Hybrid Intelligent Systems - HIS 2002* (Dec. 1-4, 2002, Santiago, Chile), international co-chair of the *6th Online Word Conference on Softcomputing in Industrial Applications - WCS6* (Sept. 2001, Word Wide Web), and has been involved on the organization of more than 10 other international conferences and workshops. Detailed information available on his CV (http://www.cec.uchile.cl/~jruizd/curriculum.html).

4 Description of team organization and effort to be spent

The creation of our robotics lab has caused great interest on the students of our engineering faculty and at many non-Faculty levels, as it was exposed in [5]. As a way of canalizing the increasing interest of our students on robotics activities, last year we decided to create two robotic soccer teams, for participating in RoboCup, from this year on. The teams are for the small-size F180 league and our team of the AIBO four-legged league. In order to achieve this difficult objective, in the four-legged league we started to work in parallel, in several fronts, with a group of 15 undergraduate and graduate students. In a first stage of "setup", one student group was in charge of the compilation of all necessary RoboCup information (official site, leagues information, rules, source codes, simulators, etc.) mainly through Internet. Another group was in charge of building our official soccer field, while a last group was in charge of buying the AIBO robots. After a hard work this stage took about 25 days.

Afterwards we started the second stage that consisted in the research work for writing the algorithms and designing the playing strategies of our team. We organized our team taking into consideration the block diagram of the final system we wanted to implement (see section 2). Our team is captained by a graduate student from our doctoral program on automation. A group of 3 to 4 students is in charge of each one of the systems blocks and a special group is in charge of making simulators. Given that we have 5 groups working in parallel, organizational aspects are very important. We have a hierarchical organization of the work and meetings every two days, whose main objective is to coordinate the work of the team. After months of very hard work, including holidays, we finished this second stage last March.

Since March 15 until the end of June we were working on the third stage, which can be characterized as "playing, playing and playing". The aim of this stage was to increase the performance of our team by means of intensive experimentation. In this context, we will participate in about one month in the next RoboCup American Open [7].

Participating in RoboCup has been an interesting challenge for us. We built a multidisciplinary team of students, which includes students with different majors like computer science, automatic control, telecommunications, electronics, power systems, signal processing and even physics. These students have been working for free and also during their vacations time. We believe that this group of students will be the basis for the consolidation of our research group on robotics. As a group we want to have the opportunity to continue demonstrating our skills in this fascinating contest.

5 Pointers to relevant publications

Our relevant publications, technical reports, as well as videos and pictures are available in our official website: www.robocup.cl.

References

[1] Chen, S., Siu, M., Vogelgesang, T., Fai Yik, T., Hengst, B., Bao Pham, S., Sammut, C.: The UNSW RoboCup 2001 Sony Legged Robot League Team. In: Birk, A., Coradeschi, S., Tadokoro, S, (eds.): RoboCup 2001: Robot Soccer World Cup V. Springer-Verlag Berlin, Germany (2002) 39-48.

[2] Guerrero, P.: Localización de un Robot Móvil Usando Información Visual Obtenida desde una Cámara Móvil. Memoria de Ingeniero Civil Electricista. Universidad de Chile, Noviembre 2003.

[3] Hornby, G.S., Fujita, M., Takamura, S., Yamamoto, T. and Hanagata, O.: Autonomous Evolution of Gaits with the Sony Quadruped Robot. Proceedings of 1999 Genetic and Evolutionary Computation Conference (GECCO). Banzhaf, Daida, Eiben, Garzon, Honavar, Jakiela, Smith, eds., Morgan Kauffmann, (1999) pp. 1297-1304.

[4] Hornby, G.S., Takamura, S., Yokono, J., Hanagata, O., Yamamoto, T. and Fujita, M.: Evolving Robutst Gaits with AIBO. IEEE International Conference on Robotics and Automation. (2000) pp. 3040-3045.

[5] Ruiz-del-Solar, J., Zagal, J.C.: How contests can foster the research activities on robotics in developing countries: Chile a case study. Accepted in International Symposium RoboCup 2003.

[6] Zagal, J.C., Ruiz-del-Solar, J., Guerrero, P., Palma, R.: Evolving Visual Object Recognition for Legged Robots. Accepted in International Symposium RoboCup 2003.

[7] http://www-2.cs.cmu.edu/~AmericanOpen03/main/

[8] http://www.robocup.cl