Vision and Control Layers of UFES_Hard_Soccer Team

Clebson J. M. de Oliveira, Julio S. Dalfior, Gilberto A. S.
Segundo, Carlos A. C. Vendramini, **Raquel F. Vassalo and
*Daniel B. Ferreira

Abstract— **The developed system was designed in three layers named respectively: vision, strategy and control. On this paper we will focus on vision and control layers because these have more technical features to explore.**
The main characteristics of the vision layer are the objects recognition based on color and the use of camshift algorithm to track objects. On the other hand, the characteristics of the control layer are the use of a ZigBee protocol-based communication device to send commands to the robots and the use of a non-linear controller to move them.

I. INTRODUCTION

Our goal on developing this system was to merge some subjects in Computer Science and Electronical Engenineering which are usually taught separately, but which are inherently linked. Thus the idea of assembling an autonomous system for the IEEE Very-Small Soccer league was a cheap way to do that. The system consists of three layers:

- Vision: this layer receives the input of the system, an image frame, and its output is an array that contains the current position, orientation and velocity of our team robots besides the position and velocity of the opponent robots and the ball;
- Strategy: from the visual information, this layer provides an output that is an array containing the desired position and orientation for each robot of our team;
- Control: this layer receives the desired pose for each teammate, calculates the suitable velocity for the motors and send this commands to the robots. This velocities will last during the system control cicle.

Figure 1 depicts the model of the system including the data flow. It's important to mention the scope of each layer to clarify which functionalities are performed by software and which ones are performed by hardware. Basically vision and strategy are software layers but the control has both software and hardware functionalities, since it includes defining by software the motor commands for moving the robots and using hardware for sending the information by a ZigBee protocol-based communication device and controlling the motors through a microcontroller.

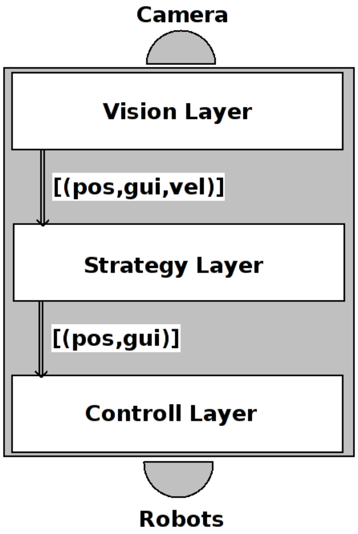

Fig. 1. Data Fluxogram

II. CONTROL LAYER

The control layer is composed by four parts:

- Control law;
- Communication device;
- Microcontroller programming;
- Hardware system;

A. *Control Law*

Our methodology was to begin using a simple controller to get, first, an overview of the system behavior and, then, improve it to a robust controller. Initially it was designed as a proportional position controller that provided good results but it did not achieve the complete goal, which was to provide a stable controller for position and orientation. This stable controller should guarantee that the robot achieves the desired pose whithout causing overhead to the strategy layer, which means that the strategy layer is responsible just for defining the pose of each robot while the control layer is responsible for executing the motor commands in order to achieve the desired configuration [1].

Currently, we are using a non-linear controller that has as inputs the current position and orientation of each robot. Then the **orientation error** and the **position error** are used by the control law to generate the commands for the motors. It's important to mention that these equations are processed

Students in Computer Science Department at Federal University of Espírito Santo, 514 Av. Fernando Ferrari, Espírito Santo, Brazil: {`cjmo208`, `juliostorch`, `gilbertosegundo`, `cvendramini`}`@inf.ufes.br`.

*Student in Eletrical Engineering Department at Federal University of Espírito Santo, 514 Av. Fernando Ferrari, Espírito Santo, Brazil: `bozi@ele.ufes.br`.

**Professor of Eletrical Engineering Department at Federal University of Espírito Santo, 514 Av. Fernando Ferrari, Espírito Santo, Brazil: `raquel@ele.ufes.br`.

in the PC and after that the motor commands are sent to the microcontroller that controls the robot.

B. Communication Device

It's used a ZigBee protocol-based communication device to send the motor commands to the robots. This specific device was chosen because it has a set up interface easy to use and a competitive price. Also this device has many different interface boards that can be bought to communicate with the ZigBee module chip, for example there are RS232 and USB interface boards. Thus it's easy to use it to make the communication between the part of the system that runs on the PC and the robots. Therefore we just have to connect a serial or a USB ZigBee interface board to the PC and stablish the RF communication with the robots.

Although the chosen board interface is very simple to use with the PC, it can not be used on top of the robots because it exceeds the robot size limits. So in order to have the RF communication module on board of each team robot, it was necessary to add the ZigBee chip into the project of the control board for the robots besides respecting the maximum size of 7.5 x 7.5 cm.

C. Microcontroller Programming

The MSP430 microcontroller was chosen because it's flexible and has a set of features that would make it easy to migrate to a more complex category like IEEE Small Size. Some of the good features of this microcontroller are that it can be programed in both C or C++ and a JTAG port can be used to load the program into the microcontroller memory. Also, after that, the loaded program can be debugged using that same port. The JTAG interface for programming and debugging is implemented through the parallel port of the PC.

D. Physical Structure and Hardware System

The team is formed by differential robots [1] made of acrylic resin donated by the team from UNESP-Bauru. Basically the structure is composed by three parts organized in layers: a box that includes two small DC engines; another section that contains four 1.2 V and one 9 V rechargeable batteries and finally the control board which is placed on the top of the structure. The system described previously can be observed in Figure 2.

The control board was designed including the digital circuit for processing and generating the control signals for the robot motors and also the RF communication system. That is based on a MSP430 microcontroller and a ZigBee module chip. The board also includes the power circuit for driving the engines. In order to avoid interference caused by noise from the motors, optical couplers were used.

The power supply for driving the motors is provided by the four 1.2V batteries, while the 9V battery supplies the digital circuit. Figure 3 shows the designed control board and the both circuits separated by the optical couplers.

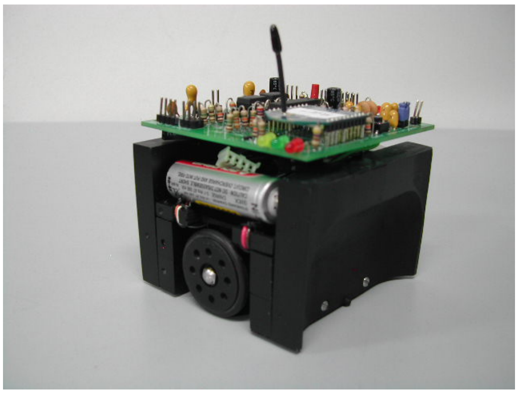

Fig. 2. Robot Structure

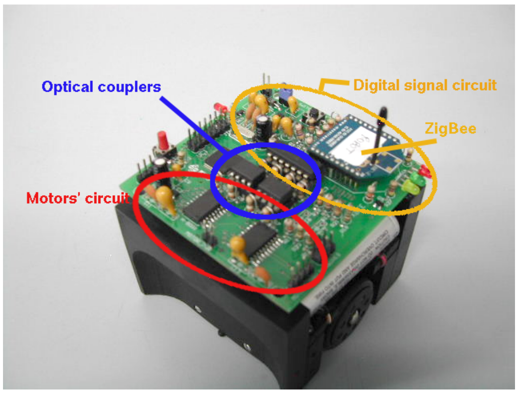

Fig. 3. The Board

III. VISION LAYER

The vision layer consists in an algorithm for color segmentation and tracking [2], [3]. Before explaining the algorithm we will list the specification of the camera that was chosen for this application:

- Maximum frame rate: 60 fps;
- Maximum image resolution at 60 fps: 640x480;
- PC interface: firewire IEEE1934a.

A. Color Segmentation

The first thing that must be done by the vision layer is to find the robots from both teams on the field. That is accomplished by a simple color segmentation algorithm that includes the following steps, which are repeated for each color used on top of the robots:

- The histogram of the color is calculated during a calibration stage, using markers with the color in different points on the field.
- The histogram backprojection onto the image is done. In other words, the color range represented by the histogram is used for marking the pixels in that range.
- Morphological operations and thresholding are done in the backprojection image.
- A labeling algorithm is used to find blobs corresponding to the segmented color.

The above steps generate an image containing all the blobs related to the different colors, but having distinct labels. The bigger blobs are considered as belonging to the robots. After knowing the colors representing our team, the two colors detected on our robots' platforms are considered to estimate their position and orientation. For the robots in the opponent team we use only one color.

To help implementing that algorithm we used the computer vision library OpenCV (*Open Computer Vision Library*) which is a free library originally developed by Intel.

B. *Tracking Algorithm Camshift*

Once the robots are detected in the first frame, their position needs to be tracked in the following frames. The same algorithm used to detect them the first time could be used in the other frames. Unfortunatelly the computational cost is too big because the whole image is scanned several times during the tracking process. To avoid that we have decided to use the Camshift algorithm implemented by OpenCV.

Given an initial search window, which is determined by the position of the blob in the last frame, it finds, inside the window and in the surrounding areas, the pixels that have colors close to the ones defined by the histogram. After that this region is segmented and its center and size are calculated and will be used to create the search window in the next frame.

IV. CONCLUSIONS AND FUTURE WORKS

A. *Conclusions*

This category is a good way to start in Robotics Championships like RoboCup and may serve as a test platform for control and computer vision algorithms. However, it can be noticed that robots' size limitations don't allow neither the usage of more sensors nor the usage of an embedded vision system. Thus if the future goal is facing more challeging applications then the ones provided by the IEEE Very-Small Soccer league it will be necessary to migrate to another category like RoboCup Small Soccer.

B. *Future Works*

The first step to improve the system described previously will be to change robots' structure to a more resistent body like one made of aluminium. The resin gears currently used suffer from a degradation process that causes gaps, affecting the spinning of the wheels and the system suitable operation.

Another improvement that can be made is to implement an embedded controller in the MSP430 microcontroller. That may turn the system more robust to gear gaps and field depressions. This can be achieved by using encoders to guarantee suitable corrections for the motors velocities.

V. ACKNOWLEDGMENTS

The implementation of the system described here was supported by Ministério da Educação of Brazil (MEC/SESU/PET) and by Fundação de Apoio à Ciência e Tecnologia do Espírito Santo (FAPES), Project N. 38425025/2007.

Specially we would like to thank professor Humberto Ferasoli, from UNESP-Bauru, for encouraging our group on moving on with many advices and through the donation of structures for three robots. Also we thank professor Reinaldo A. C. Bianchi from Centro Universitário FEI for the good discussion on computer vision during a visit to UFES.

REFERENCES

[1] G. Dudek and M. Jenkin, Computational Principles of Mobile Robotics, Cambridge University Press; 2000.

[2] R. Jain, R. Kasturi and B. G. Schunck, Machine Vision, McGraw-Hill; 1995.

[3] B. Klaus and P. Horn, Robot Vision, The MIT Press; 1986.