Lomu, an Autonomous Mobile Robot with Robust Architecture and Components

Nicolas Uebelhart, Florian Glardon and Pierre-François Gauthey

HESSO-HEIG, University of Applied Sciences of Western Switzerland,
Yverdon-les-Bains, Switzerland,
{Nicolas.Uebelhart, Florian.Glardon, Pierre-Francois.Gauthey}@eivd.ch

Abstract

The Laboratory for Robotics and Automation (LaRA) at HESSO-HEIG has developed several autonomous mobile robots with original architecture and control structure. These robots have participated to numerous EUROBOT competitions at Swiss and/or European levels. This paper describes a mobile autonomous robot named "Lomu", which is the most recent offspring of our design. The building blocks of Lomu are, as much as possible, made of industrial off-the-shelf components. Other key features are the excellence and modularity of software and hardware, technical functionality and operational reliability.

1. INTRODUCTION

The Laboratory of Robotics and Automation (LaRA) of the HESSO-HEIG is active since many years in the field of mobile robotics and has developed several autonomous mobile robots comprising an original architecture and a particular control structure. These robots have taken part in many contests of EUROBOT at the Swiss and/or European levels. This paper introduces the autonomous mobile robot “Lomu”.

The first section of this paper describes the operation of the mobile autonomous robot “Lomu” and the various tasks which it carries out. The second section describes the hardware architecture of “Lomu” and its various components. Finally, a third section describes the control of the components and the programming of “Lomu”.

2. THE AUTONOMOUS MOBILE ROBOT “LOMU”

The robot “Lomu” was developed to take part in the contest of EUROBOT 04, the aim of which was to score a maximum of points in 90 seconds by placing small rugby balls in the adversary zone or by shooting them into the opposite goal. The robot was to be able to avoid obstacles placed randomly at the beginning of each match as well as contacts with others robots.

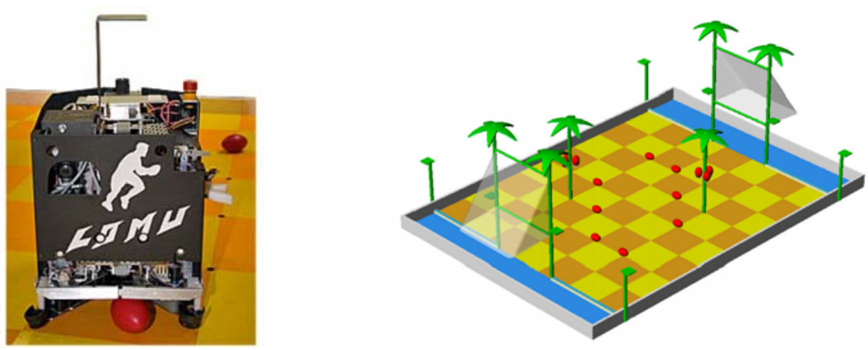

Figure 1. The robot “Lomu” and the EUROBOT 04 table

2.1 CATCHING THE BALLS

By means of an intake guide and a system of belts, the balls are conveyed successively into the gun. The robot can contain a maximum of three balls in its charger during its displacements towards the shooting zones. Luminous barriers make it possible to facilitate the storage and the ball counting.

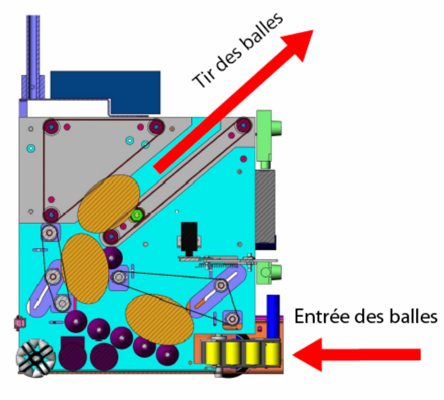

Figure 2. Cross-section of Lomu, showing the catching ball system and shooting system

The Balls are positioned on supports (coconut-trees) placed randomly on the playing table at the beginning of each match. The balls are catches by a hinged arm equipped with an ejector finger positioned on the left side. The finger ejects the balls on the top of the robot; they are then swallowed by the gun and finally stored in the charger.

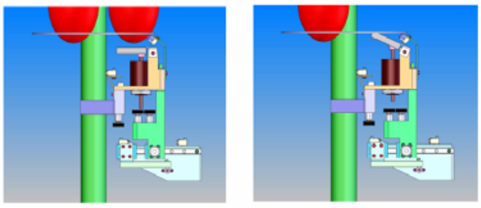

Figure 3. Finger eject system

2.2 AVOIDING OBSTACLES

A video camera takes an image of the ground at the beginning of each match to locate the positions of the coconut trees. Four ultrasound sensors at the front and a series of contact sensor at the front and at the back of the robot make it possible to detect the other robots, the edges of the playing table as well as the coconut tree in the event of camera detection error.

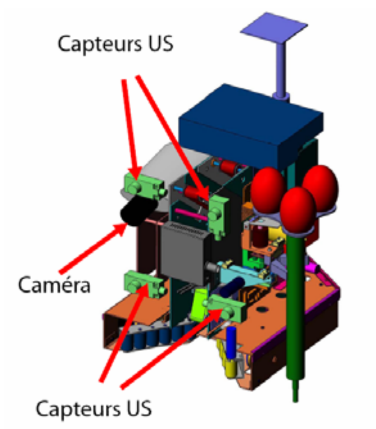

Figure 4. Video camera and sensors

3. THE ARCHITECTURE

The internal communications are based on an Ethernet network [1].

The architecture of the robot is as follows:

- A HUB. which connects physically the various components;
- A compact notebook computer, which is our decision support centre;
- A specialized motion controller with trajectory interpolation;
- A video camera;
- An Inputs/Outputs Controller.

The flexibility of this architecture makes it possible, if necessary, to replace an element by another, to add or remove elements without an advanced study of their integration [6].

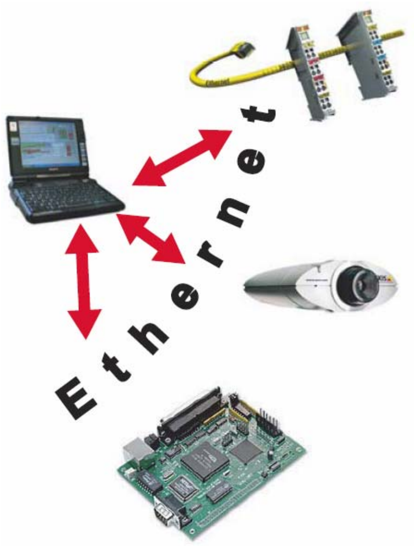

Figure 5. The hardware architecture

3.1 THE COMPACT NOTEBOOK COMPUTER

The computer is the base of the system. It has the function to manage the whole of the system and to process the data coming from the other elements of the network. Its function is also to control the data transfer on the network. The selected PC is a very compact notebook running on Windows XP professional.

3.2 THE HUB

The purpose of the hub is to connect physically the various elements of the network in a "star" mode. With a hub, all elements have access to all information at 100Mbit/s.

To communicate properly, the data packets are supplemented by the IP addresses of the receiving and transmitting devices. On the network, when the hub receives a packet to be transmitted, it will send this information to the various parts of the network until it finds the destination device. The various elements will look at all information, but will retain only what is addressed to them. In this way, each part of the network uses only the information which is intended to it.

3.3 THE MOTION CONTROLLER

The motion controller, made by GALIL, can control two axes independently. This board can be connected to the Ethernet port. The fact that movements can be programmed directly on the board memory makes it possible to offload the decision supporting task, which can therefore send simple motion commands.

3.4 THE CAMERA

The camera used is a network camera AXIS 2100. It comprises an RJ 45 connector to connect it directly on the network. It takes up to 10 colour images per seconds according to the ambient luminosity and has an automatic adjustment of gains. The chosen picture resolution is 320x240 pixels. A wide-angle lens (eye-fish) was added to have the full-size view of the playing table.

3.5 THE INPUTS/OUTPUTS CONTROLLER

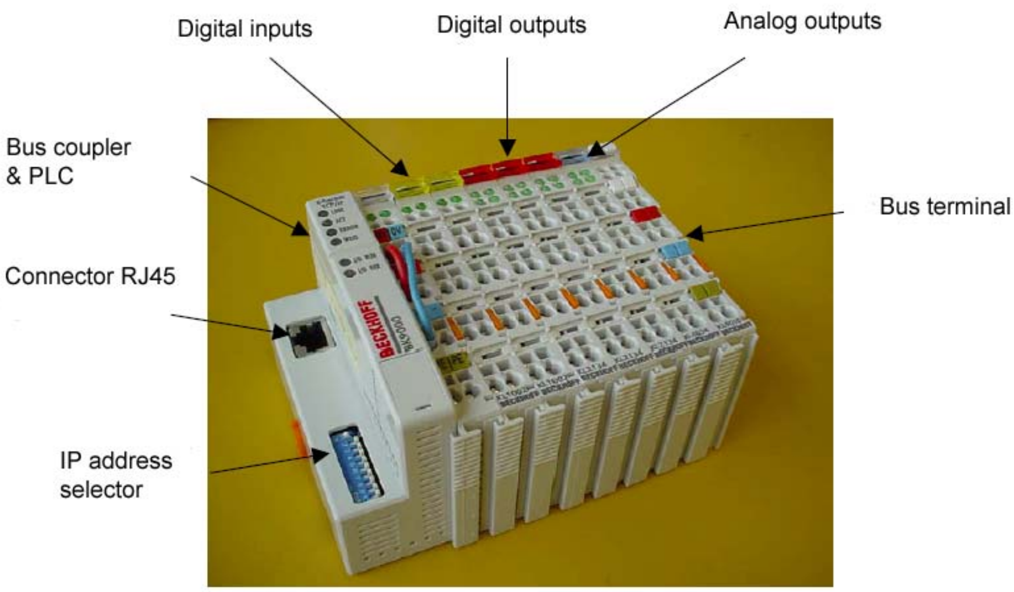

Figure 6. The inputs/outputs controller BC900 made by BECKHOFF

The inputs/outputs controller is a BECKHOFF BC9000 model. This element is fitted with screw terminals to connect the various inputs/outputs. The choice of these terminals is very large and makes it possible to build a unit according to each need. There are analogue or digital inputs/outputs corresponding to different voltages, such as 0 - 10V, 24V, 220V... It is also possible to choose a number of inputs/outputs necessary for a project. For the Ethernet communications the BC9000 is equipped with an integrated server which has a fixed IP address. The management of low level information between the terminals is done under TCP/IP protocol by a MODBUS layer, functioning according to the standard 1131. An internal module (PLC) can be programmed to execute tasks locally [4].

3.6 THE SENSORS

The robot comprises two types of sensors:

- The interoceptive sensors which give information relative to the robot.
- The exteroceptive sensors which give information about the external environment.

3.6.1 THE ODOMETERS

The odometric data are obtained with coders. The coders send the odometric signals allowing, after a simple transformation, to have the distance covered by a wheel. Knowing the distance covered by each wheel, one can calculate the displacement carried out by the robot [2].

3.6.2 THE ULTRASOUND SENSORS

These sensors are analog and make it possible to obtain distance information from a likely obstacle.

3.6.3 THE BUMPERS

The bumpers make it possible to detect collisions. The switches are closed (N.O.) or open (N.C.) when there is contact between the bumper and an obstacle. Information that they return is of type "all or nothing" (ON-OFF type).

3.6.4 THE OPTICAL SENSORS

The optical sensors are luminous barriers which make it possible to detect and to place correctly the balls in the charger of the robot. They are sensors "all or nothing" (ON-OFF type).

4. ROBOT'S DISPLACEMENT METHOD, CONTROL AND POGRAMMING

The Lomu's technique of displacement is like robotics industrial arm. This concept allows limit the complexity of displacement methodology using three virtual joints (X;Y;θ) with the Denavit and Hartenberg method.

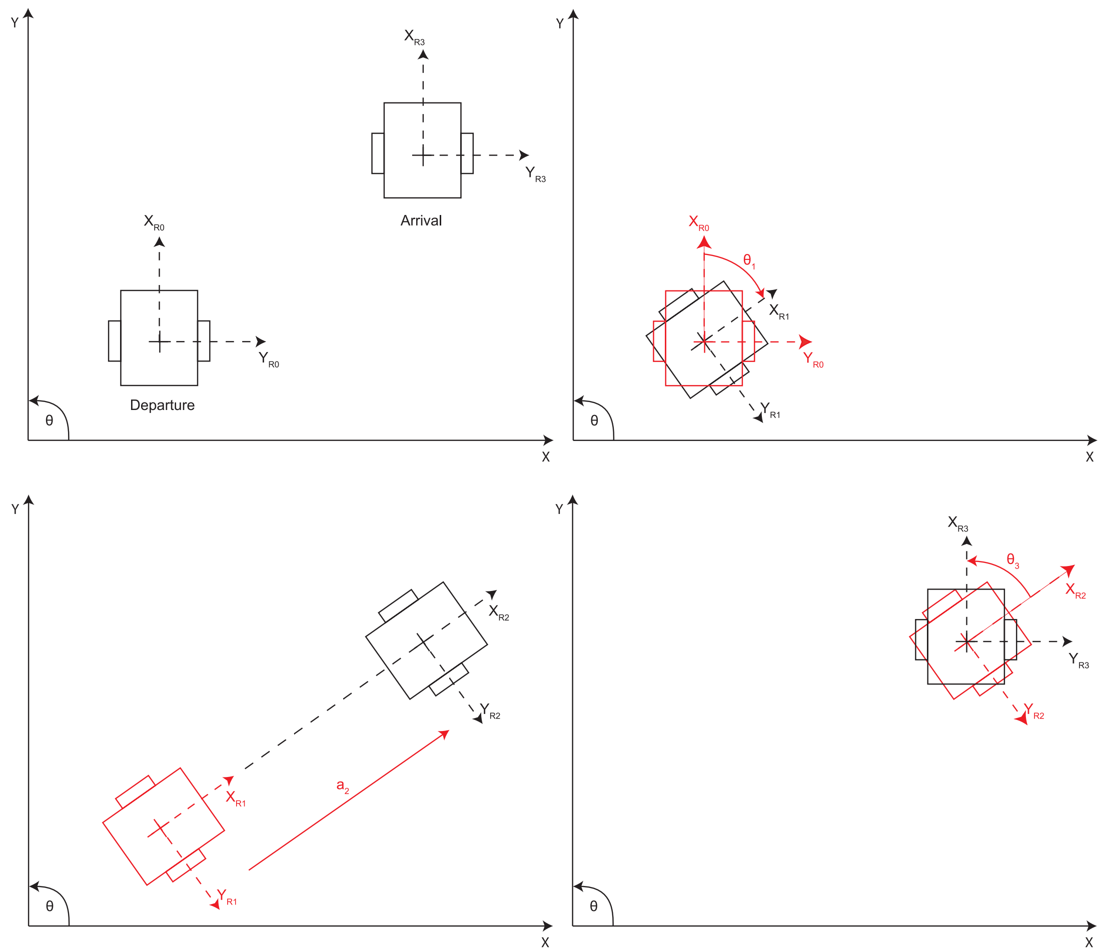

Figure 6. Robot's technique of displacement

.

$$A_0^1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_1^2 = \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2^3 = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation (0-1) Translation (1-2) Rotation (3-2)

The final matrice of robot's position is:

$$\boxed{A_0^3 = A_0^1 \cdot A_1^2 \cdot A_2^3}$$

The programming of the robot and the trajectory management are carried out in a context “Piaget” multi-agents real time implemented in C++ language. “Piaget” facilitates the programming of the strategy and makes it possible to quickly carry out modifications between the matches with primitive instructions of the type:

```
2024: ReculerAGN(30);              //backward 30 cm (relative position)
            break; case

2025: SetTransAGN(30,90,90);       //Set  the translation at 30;90;90
            break; case            //Absolute coordinates (x;y;φ)

2026: MoveAGN(Trans);              //Go to the position define by “SetTransAGN”
            break; case

2027: SetTransAGN(30,240,0);       //…
            break; case

2028: MoveAGN(Trans);              //…
            break; case

2029: SetTransAGN(105,240,90);     //…
            break; case

2030: MoveAGN(Trans);              //…
            break; case

2031: GoState(2021);               //Go to the state 2021
            break; case
```

The structure of the program is composed of various parts dedicated to different tasks.

```
while (! DesiredInteraction)
{
Ticks+=1;
Task01();
Task02();         // Move on step
Task03();         // Read keyboard
Task04();         // Step by Step Movements
Task05();         // Strategy
Task06();         // Inputs/Outputs
Task07();         // Displays
Task08();         // Movements spatial
Task09();  // LED management
Task10();         // Analyse images
Task11();         // Finger reflex
Task12();         // balls management
Task13();         // Inputs test
Task18();         // Interpreter “Piaget”
}
```

The various programmed tasks are checked every 100 nanoseconds on average.

The purpose of the Analyse image task is to process the data coming from the camera AXIS 2100. The image processing is carried out by various processes [3]:

- Average filter
- Median filter
- The specific modes of processing

The specific modes of processing make it possible to define the attributes which are characteristic of a colour to be isolated. For the Swiss robotic cup 2004, several modes were defined:

- The red mode to detect the balls
- The green mode to locate the coconut tree
- The magenta and blue modes to locate the luminous markers on each side of the opposite goal.

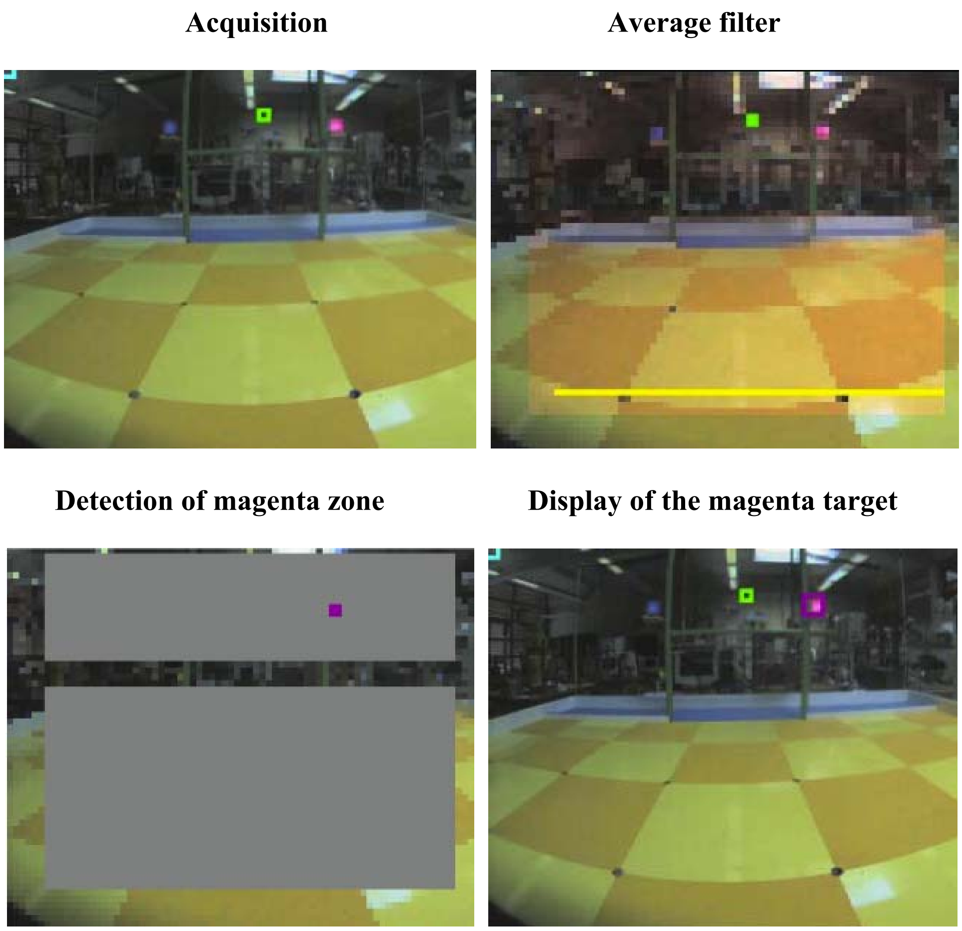

Figure 7. Image processing

Figure 8. Main control panel and image processing control panel

The Main control panel allows simulating the displacement of the robot, to have a visual monitoring of the inputs/outputs, to regulate speed and acceleration and check the correct operation of the program. The control panels of the video camera allow to regulate the gains and to define the position of the target on the image which allows the correlation between the distance on the playing table and the image pixels.

4. CONCLUSION

The particular architecture of the autonomous mobile robots of the Laboratory of **R**obotics and **A**utomation (LaRA) of the HESSO-HEIG proved its effectiveness not only regarding teaching at the laboratory but also for the Swiss and European robotics cups (EUROBOT). This paper showed the effectiveness of this architecture by the fact that it employs industrial components which can be replaced without complicated modifications for their integration. The various sensors that “Lomu” uses make it possible to obtain a great safety during its displacements regarding obstacle avoidance and localization. The structure of the programming “Piaget” makes it possible to carry out modifications quickly and clearly facilitates programming of strategies.

REFERENCES

[1] S. Charmillot, Diploma work, "Realisation of autonomous mobile robot with internal communication with Ethernet", Diploma report, HESSO-HEIG, Yverdon-les-Bains, Switzerland, December 2002.

[2] M. Dellatre, "Positioning and avoidance of obstacles for autonomous mobile robots", Diploma report, HESSO-HEIG, Yverdon-les-Bains, Switzerland, September 2004.

[3] S.Von Wyl, "Image processing bidimentional ", HESSO-HEIG, Yverdon-les- bains, Switzerland, July 2004.

[4] J. Luethi, "PLC management objects", HESSO-HEIG, Yverdon-les-bains, Switzerland, February 2004.

[5] J.-D. Dessimoz, "Is a Robot that can Autonomously Play Soccer Intelligent?", Proc. Intern. Conf. on Intelligent Autonomous Systems, No 6, E. Pagello et al (Eds), IOS Press, co-sponsored by IEEE, Venice, Italy, July 2000, pp. 951-958

[6] Antoine Trad, Jean-Daniel Dessimoz, “An Architectural Design Pattern Proposal for Mobile Robot Systems”, Conference proceedings of the International IMS (Intelligent Manufacturing Systems) Forum 2004, held in Cernobbio - Italy on 17th-19th May.pp. 346-351,(http://www.imsforum2004.org)