A New Modular Architecture for the Mobile Robot MORDUC: from the Hardware to the SLAM Algorithm

Daniele Caltabiano[1] and Giovanni Muscato[2]

[1] University of Catania, Catania, Italy, daniele.caltabiano@diees.unict.it
[2] University of Catania, Catania, Italy, gmuscato@diees.unict.it

Abstract

This paper describes a new modular architecture for mobile robots that involves the whole system, from the low level hardware to the high level software. The hardware is composed only by two physical parts: the main computer and the sensors turret, the latter is composed by various boards, each one adopted to interface a different kind of sensors, assembled together in a robust manner and self connected using a single bus. The main advantage of this system is that the connection of a new kind of sensor implies only to add a new board in the turret. The Software is divided into two levels: the low level module realizes an abstraction layer to the sensors; moreover a simple navigation algorithm has been implemented in this level to avoid obstacles even if the robot is tele-operated. The high level module is used to tele-operate the robot (GUI) and to implement the SLAM algorithm, it retrieves the sensor data via RPC and can physically reside on a different PC connected to the same LAN (wired or wireless).

1. INTRODUCTION

In very active robotic research laboratories, like in the universities, it is very important to have a modular platform to try new sensors in an easy manner, without the need to redesign each time the whole hardware.

At the same time, a similar problem involves the software design: for battery autonomy, time constraint and laboratory space reasons, generally a real test on a mobile robot is very complex, a modular approach on the software would permit to use alternatively the real robot or a simulated one in the preliminary design phase and proceed with the real test only in the last stage. Moreover the robot hardware is unique while many different instances of a simulated robot can be studied at the same time.

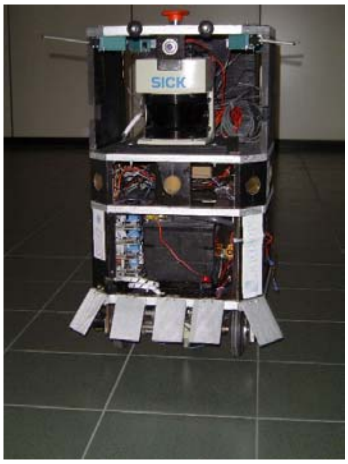

Figure 1. The robot MORDUC

For these reasons, the mobile robot MORDUC, "Mobile Robot DIEES University of Catania", has been recently redesigned form the Hardware to the Software point of view [1].

The next section examines the overall structure of the robot MORDUC; the third and fourth section show respectively the realized hardware and software; the last section reports the conclusion of this work.

2. THE ROBOT MORDUC

The MORDUC system is a differential drive robot with a rear castor wheel for indoor environments. It is composed by a wood chassis that divides the robot into four parts. The lowest part carries the batteries, most of the electronics and the motors, the second level is occupied by a ultrasound sonar belt, in the third level a laser scanner and a webcam are positioned on the front side, while the upper level is used to carry a laptop computer.

The PSU (Power Supply Unit) is composed by two sealed Lead-Acid Batteries connected in series, the main one (50Ah) is used to power the motors and all the electronics, the second one (7Ah) is used to generate the 24V needed to the Laser scanner.

Some contact switches are placed on the front side and on the external border of the robot in order to prevent dangerous crashes.

An emergency button is placed on the top to stop the robot immediately.

The robot has been used for various researches concerning the problems of mapping the environment and autonomous navigation.

3. HARDWARE

The Hardware of the robot has been redesigned in order to be easily updated for interfacing new sensors and at the same time to be robust and compact. The electronic boards are of fixed size (10cm x 10cm) and are assembled in a small turret throughout four small bolts. A communication bus has been designed to facilitate the connections: both the power supply and the communication interfaces (I^2C, CAN, RS422) are part of the same bus, in this way no cable is needed between the boards and a single cable is needed to connect all the boards of the turret to the PC.

Fig. 2 shows the hardware of the robot MORDUC and where it is located upon the robot.

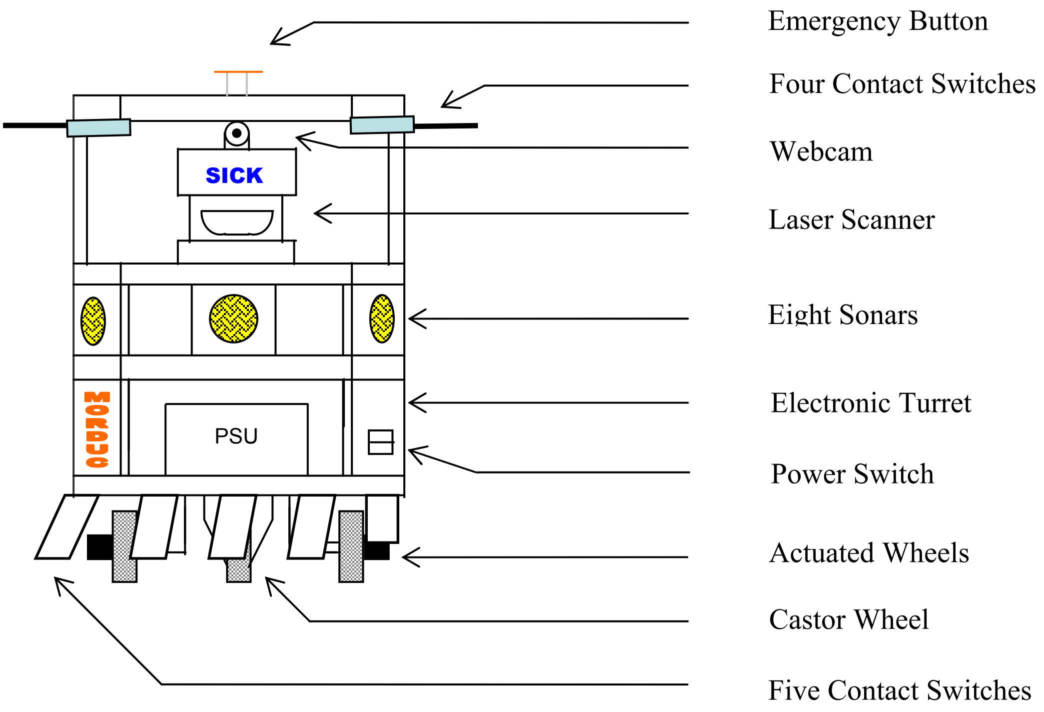

Figure 2. The Hardware of the robot MORDUC.

In Fig. 3 a picture of the hardware turret is shown. It is composed by five boards: the first one, mounted on the top for heat dissipation purposes, is the *power supply board* and provides the regulated 5V-3A to the entire turret. The second board is the *PC-interface*, used to connect the turret to a PC throughout three possible connectors: a DB9-F is used to connect the PC RS232 to the RS422 bus of the turret, a DB25-M connects the PC parallel port to the I^2C bus and a DB9-M is used to interface the CAN bus.

Figure 3. The hardware *turret*

Actually the electronic turret of the robot MORDUC is connected using the I^2C interface, even if some of the boards are compatible with the other two standards implemented on the bus (CAN and RS422).

Unfortunately the pre-existing hardware can not be placed in the turret and must be connected using other cables, this is the case of the power boards MD03, managed directly from the laptop via the I^2C bus and the eight sonar ranging modules (Polaroid 6500) connected to the third board of the turret. This board, controlled via I^2C interface, is used to measure the time of flight of ultrasonic pulses emitted using an electrostatic transducer and echoed back by the obstacles.

The fourth board is used to decode the quadrature signals of the front wheels encoders, the motor control algorithm is implemented on the laptop that periodically (20Hz) acquire the encoder signals from this board via the I^2C bus and then set the motor speeds in the Power board using the same bus.

The last board, *Expander*, is a collection of I^2C devices: 32 digital I/O, 8 analog inputs, 2 analog outputs, a 512Kb serial EEPROM and a temperature sensor. This board is used to monitor the environmental temperature to correct the sonar readings and to interface the contact switches (there are five contact switches at the front-lower part of the robot and four around the upper part).

All the described boards have some dip-switches to configure the device address on the bus in order to simplify the connection of supplementary hardware.

The hardware is completed by a webcam and a Sick Laser Scanner connected directly to the laptop by using respectively a USB and a serial port. The Laser scanner provides a 180° scansion of the environment with a resolution of 0.5° and 1 cm.

4. SOFTWARE

The software has been divided into two parts which communicate throughout a Wireless LAN by using Remote Procedure Calls (RPC). The low level software, called *MORDUC System*, implements an abstraction layer with the sensors and stores all the data in a database in the laptop RAM. The high level software, called *RemoteClient*, is executed on a different PC and is responsible for the tele-operation GUI (Graphical User Interface) and for the SLAM (Simultaneous Localization And Mapping) algorithm [2]. A simple obstacle avoidance algorithm has been implemented in the MORDUC System in order to prevent crashes when the wireless communication with the GUI has large delays [3].

This solution have several advantages: first of all, any change on the GUI or the SLAM algorithm can not compromise the stability of the system, since it is executed on a different machine; second, the computation of the SLAM algorithm and GUI are not loaded on the onboard PC and third the MORDUC system can be substituted with a simulator in a transparent way.

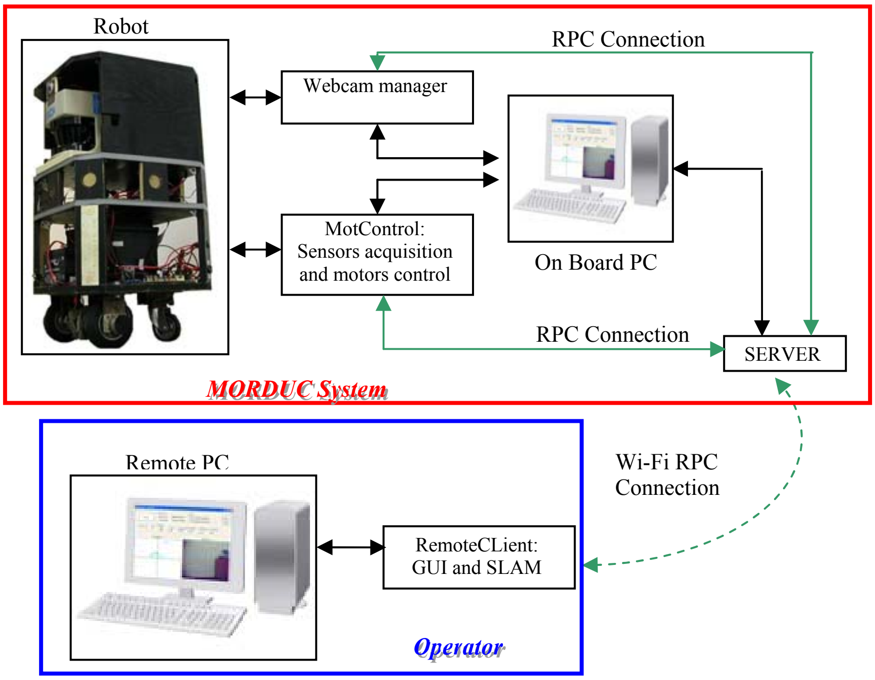

Figure. 4. The architecture of the software.

4.1. MORDUC SYSTEM

The MORDUC System is composed by three programs: the *Server* retrieves all the sensors data from the other programs and stores them in a local database; the *Webcam Manager* takes pictures from the webcam and send them to the server; the *MotControl* is the heart of the system and is used to interface all the sensors and to control the motors.

A shared variable (command) is available on the server to allow the user on the GUI to choose which sensor must be acquired and which one must be used for the low level obstacle avoidance. In particular, when the system is started, the *Webcam Manager* does not start the webcam, but periodically reads commands from the server. The user can hence choose from the GUI to use the webcam and if another webcam is connected on the operator PC, its video stream can be shown in the monitor of the robot by the *Webcam Manager*, to achieve a tele-presence system.

The *MotControl* has the same behaviour, it periodically reads commands from the server, the operator can set the robot linear and angular speed of the robot using a joystick, and then these commands are transformed in motor reference speed in the *MotControl*. The user can also choose to use the sonars and/or the laser scanner, in these cases the *MotControl* will sample the corresponding board or sensor periodically and send the measures to the server.

As mentioned before the *MotControl* implement also a low level obstacle avoidance algorithm, it is a simple check on the acquired sensor, if an obstacle is in the path of the robot, the linear speed is set to zero and only rotations are allowed. Three kind of sensors can be used for this purpose: the sonar belt, a minimum distance of 50 cm from the surrounding obstacles is kept; the laser scanner, in this case since the sensor resolution is much higher, a minimum distance of 20cm is allowed both on the front and side part of the robot; the nine contact switches are used to stop immediately the robot in case of a collision.

A simulator of the MORDUC system has been realized using VRML; a simple GUI permits to create the environment selecting vertical walls of fixed height. The simulator is totally compatible with the real MORDUC robot, since all its main sensors have been implemented, i.e. encoders, sonars and laser scanner. In this way the *RemoteClient* can be used indistinguishably with the real robot or the simulator.

Figure 5. Example of the simulated environment.

4.2. REMOTECLIENT: GUI AND SLAM.

The GUI is organized in a single big form. The upper part of the interface is dedicated to the settings: the user can select which sensors must be turned on, which one must be used for the low level obstacle avoidance, described in the previous section, and to activate the video streaming in both directions. The robot webcam stream can be shown in the *RemoteClient* and eventually a webcam connected to the *RemotePC* can be shown in the *Webcam Manager interface* onboard the robot (tele-presence).

A part of the interface is reserved for the monitoring of the batteries voltages, the state of the switches and the temperature of the environment. Another part is used for the graphical output of the sonars and laser scanner. The laser scanner readings are also elaborated using the *split and merge* algorithm. The detected walls are plotted on the same graph using a light blue colour. The reconstructed map is shown in real-time in a dedicated frame, where the robot is shown always in the centre with the actual orientation. Some checkboxes are used to enable the visualization of these outputs in order to reduce the computational requirements.

Fig. 6 shows a picture of the *RemoteClient* during the reconstruction of an indoor environment.

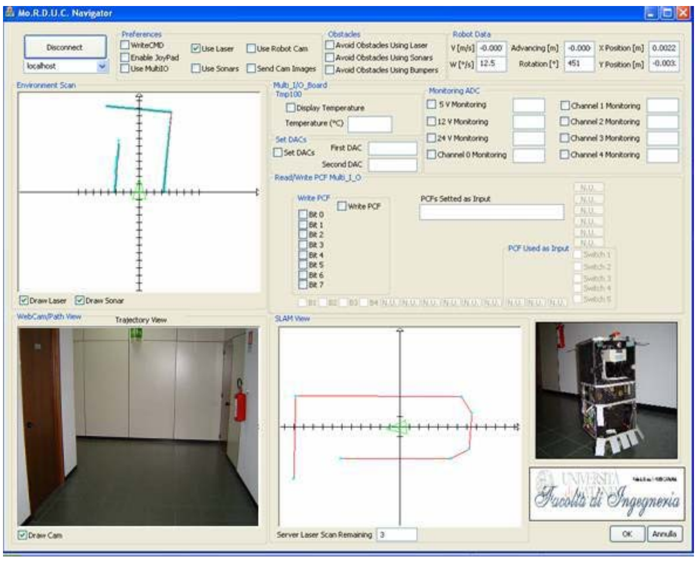

Figure 6. Example of the RemoteClient interface.

The SLAM algorithm is based on an Extended Kalman Filter (EKF) that is used to merge the local maps in a large metric map. The local maps are obtained from the laser readings with the split and merge feature extraction algorithm, where the extracted features are the wall vertexes. An estimation of the position of the features is obtained in the prediction phase of the EKF using the odometry.

A simple autonomous navigation algorithm (called *Decision*) has been also implemented in order to test the simulated system: this algorithm decides the desired path calculating the position with less obstacles in the laser readings.

5. CONCLUSION

In this paper a new modular architecture for mobile robots has been described both from the hardware and from the software point of view.

The hardware has been organized in a compact and robust small turret (10cm x 10cm), where a single 50pins bus connects all the boards, providing them the needed power and the communication interfaces with the onboard PC.

The hardware is easily scalable: a more complex system can be controlled by simply connecting more boards to the turret.

The motor control is actually done by the onboard computer at 20Hz, a powerful 3-axes motor control board, based on the microcontroller ST10 and compatible with the turret, has been already realized and work is in progress to integrate it on the MORDUC System.

The software has been divided into three main parts: the first part is used to acquire the sensors and to control the robot at low level, the second part is used to control the robot at high level and the Server permits to exchange information between the other two programs.

A robot simulator has also been implemented together with a simple autonomous navigation algorithm. Fig. 7 shows the interoperability between these functional blocks; in the centre of the figure, the server is always present and permit to connect a block of the left side with one of the right.

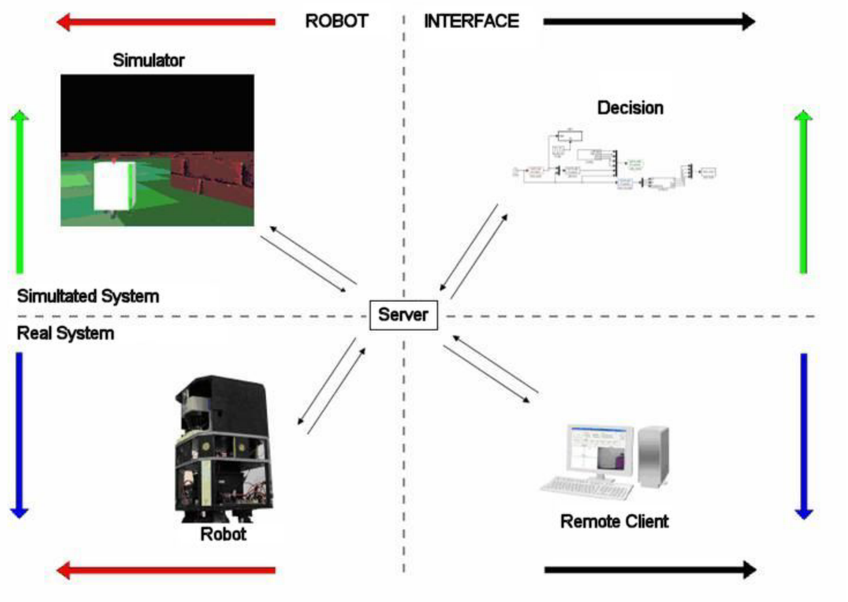

Figure 7. MORDUC System – Simulator interoperability.

REFERENCES

[1] G. Muscato, G. Scalia, "MORDUC: An INTERNET connected mobile robot", CLAWAR 99 International conference on climbing and Walking Robots, Portsmouth, U.K. , 14-15 September 1999.

[2] J. A. Castellanos, J. T. Tardos, "MOBILE ROBOT LOCALIZATION AND MAP BUILDING, A Multisensor Fusion Approach", Kluwer Academic Publishers, 1999.

[3] R. Siegwart, I. R. Nourbakhsh, "Introduction to Autonomous Mobile Robots", MIT Press, 2004.

[4] H.R. Everett, "Sensors for Mobile Robots: Theory and Application",AK Peters Ltd, 1995.