Abstract: This paper deals with the general problem of designing hardware architectures for mobile robots. In particular, four different architectures of mobile robots have been studied. An architecture for an industrial forklift, used in the automation of management and transport processes is firstly described. An architecture for a convoying application with an electric car is secondly showed. Finally, two RT platforms are described based on distributed architectures mainly using CAN field bus. The Robuter II and the PCBot II vehicles, have been used for research applications in these latest implementations.

Key-Words: Hardware Architectures, Mobile Robots, CAN field bus, RT O.S.

1 Introduction

Robots in industry have released man to carry out tedious, and in most of the cases, dangerous tasks. Despite of this, first robots were only able to operate with elements around its scope. This difficulty was rapidly overcome by introducing automated transport systems which bring pieces to the robot (conveyor belt) or move the robot near the pieces (rail). Nevertheless, traditional transport systems introduce stiffness routes which following fixed and predetermined paths. Small changes on the routes forced to an important redesigning of the system.

Therefore, quickly changes on demand forced to substitute conventional transport systems for automated guided vehicles (AGV). The AGVs are able to follow pre-established routes on the floor like magnetic strip, painted lines, etc. Nevertheless, they present a large number of limitations, such as to compute optimal paths or to avoid obstacles.

In these cases, they are usually provided with sensors to detect obstacles but stop until they are removed. Traditional AGVs did not have desired flexibility degree and therefore, any modification of the guided system implied a high cost investment.

Nowadays, the state of the technology allows a complete range of sensors. Despite of this, autonomous vehicles require powerful control and sensor perception in order to depict some kind of intelligence. Specific control algorithms are being used on industry processes being deeply studied on research centres and universities [1] [2] [3] [5].

Based on application requirements an appropriate hardware and software architecture have to be defined being able to interconnect sensors and actuators.

In this sense, the purpose of this paper is to describe several hardware architectures implemented for four different types of mobile robots. Some analysis of the proposed hardware architectures are included as well as indications about the software structure.

2 Hardware Architectures applied to Mobile Robots

This section is intended to describe different hardware and software architectures applied to mobile robots. It is interesting to remark that any hardware and software architecture should be particularized for a given mobile robot and different solutions can be efficiently implemented on different platforms. The following evaluation criteria can be applied for selecting the most appropriate architecture:

- Modularity.
- Ease of implementation.
- Platform portability.
- Robustness.
- Solution cost.
In this study we consider four mobile robots\(^1\) used for different applications. We firstly propose a hardware and software architecture for an industrial forklift to be used in the automation of transport processes. A proposed architecture for a convoying application with an electric car is secondly showed. Then, a RT platform is proposed for the commercial robot Robuter II, and an architecture is described for a small differential robot, named PCBot II, both used for research applications.

### 2.1 Architecture for an Industrial Forklift

An industrial forklift is used for automation of management and transport processes within a plant. The main objectives of transport automation are:

- Interact with the management process through a monitor installed on the vehicle.
- Move from one point to another of the plant, according to pallet transport requirements.
- Obstacle avoidance and speed regulation, when objects cross near the vehicle.
- High security level defining a protection area around the vehicle.

An industrial forklift has been completely automated by introducing additional sensing and a controller system based on a PLC. The vehicle is composed by three DC motors driving three wheels, two at the front and the other at the rear part of the vehicle. An accelerator pedal commands an electronic differential which governs the two front wheels synchronising them to avoid sliding. Two incremental encoders, one at each wheel are used for speed measurement. A gear establishes vehicle’s sense (forward, neutral or reverse sense).

This vehicle has a DC motor driving the steering wheel with an absolute encoder coupled to the shaft for angular measurement. It is interesting to remark that this angle affects also to the speed of the front wheels.

Two laser rangers, PLS from SICK, have been installed, one at the front and the other at the rear part of the vehicle. These sensors activate two digital inputs when an intrusion of two configurable fields occurs: warning and protection. Laser sensors also provide range information of a 180° scan with 0.5° and 5cm precisions via RS232 with a compatible PC baud rate up to 38400 bauds, taking approximately 250ms time delay on data request. These scanner lasers can be also connected to an interface unit PLI, which can communicate with them at 500000 baud rate. It also provides an interesting functionality, which can generate 8 different protection zones for vehicle’s speed regulation, activating a digital output for each zone. A ring of 6 US and 8 IR sensors has been distributed around the vehicle in order to cover lateral zones of the vehicle.

IR sensors, GP2D02, transmit distance information up to 70cm. The physical principle of this sensor is the inclination of the IR light bounced on objects. Furthermore, a study made with different coloured objects demonstrated that in the worst case the precision was of ±1cm, even with natural light.

US sensors from Polaroid 600-Series are used to detect object presence up to 8m with a opening angular range of 30°. The ultrasonic wave is transmitted and received by the same transducer, therefore it is required a waiting time interval between the end of the transmission to the beginning of the reception in order to avoid false echoes. An amplified 50Khz burst pulse signal generated by a µController is send to the transducer. The received signal is amplified by the sonar receptor TL852. The power of the ultrasonic wave weaken for large distances, nevertheless, TL852 is able to keep sensibility for all the operating range. It is interesting to remark that, distance measurement is based on the principle of time-on-the-fly (TOF).

Each IR and US sensor is integrated in a CAN node composed by a PIC16F877 µController with MCP2510 and 82C250 as CAN controller and driver respectively. The CAN controller is connected to the µController via the SPI bus. All the CAN nodes are connected through a CAN bus ring communicating at 500Kbauds.

The described ranger sensors are used as protection mechanism and for data fusion and map building algorithms [4].

Every CAN node has an own higher level protocol. On power connection the node transmits an “I am alive” message indicating that the node is waiting for configuration. This message is unique for each node and its priority depends on the task. Exist three configuration mode: in the first one the node interrupt its task and waits for a new configuration mode; in the second one data is send by request. It also exists a message for resetting all the CAN nodes and another for synchronizing all of them when the nodes are working in periodic mode.

---

\(^1\) Robots are located in “Laboratorio de Robótica Móvil. Escuela Universitaria de Ingenieros Técnicos Industriales. Universidad Politécnica de Valencia”. Spain.
In this industrial vehicle, real-time and robust control requirements are implemented by closing the control loop with an industrial PLC, managing sensors and actuators. In a higher level an industrial computer communicates with the PLC. It also handles with laser ranger via RS232 and with IR and US sensor through CAN field bus, as shown in Fig. 1.

A client application running on the on-board computer allows to communicate (wireless) with a server application for data monitoring. In this sense, multiple industrial vehicles can be automated within the plant, running all of them as client of a server application running on a main computer.

It is clear that the main criteria considered for the selection of this architecture has been the robustness of the PLC. On the other side, with the industrial computer is feasible to implement object detection and avoidance algorithms [2], data fusion and map building applications [4] and multi-rate control algorithms [5] due to the fact that sensors work at different sampling periods, ie: laser and US sensors.

The high hardware architecture cost using a PLC is completely justified on an industrial environment, where the security and control are extremity important. Furthermore, it is interesting to remark that PLC have: a whole set of I/O modules for connecting sensors and actuators without additional electronic hardware; advanced instructions (floating point, control flow, table management, etc...); enough program memory and data memory for this application; interrupt tasks up to 1ms; high speed instruction execution, etc...

2.2 Architecture for an Electric Car

An electric car is used for convoying applications, where the speed of the vehicle must be regulated in accordance to the speed of the front vehicle.

The hardware architecture is based on CAN field bus, due to the distributed location of sensors and actuators around the vehicle. Each sensor or actuator is connected to a CAN node, similar to the nodes previously described, with the same configuration modes.

The electric motor is governed by the accelerator pedal, through a potentiometer. So far, the breaks are mechanical and the speed slow down by reversing the polarity of the motor. Nevertheless, a potentiometer on the breaks disables the accelerator pedal and can be used for break monitoring. Neutral, forward and reverse movements are governed by a selector, where a change on sense is not produced until the speed goes to zero. The speed of the vehicle is measured with a Hall effect sensor coupled to the wheel.

The speed control loop is closed with: a D/A converter CAN-node, substituting the
potentiometer signal; and with a Hall effect CAN-node, which measures the vehicle’s speed.

The steering wheel has been automated by introducing a power-assisted steering. Three CAN-nodes has been designed in order to control the steering wheel position: the DC motor driving the power-assisted steering is controlled through a D/A CAN-node; the steering wheel position is measured with an absolute encoder CAN-node; and the torque applied to the power-assisted steering is measured to close the torque loop.

A scanner laser is used for measuring distances to the front vehicle. This sensor has as main interesting functionalities: 180° scan measurements with 0.5° angular precision, the shortest distance and intrusion detection in three configurable fields. A laser CAN-node has been designed to treat laser information and to transmit relevant data through the field bus. The vehicle is also equipped with a ring of 16 US and 12 IR sensors used as medium and short distance protection mechanisms respectively.

Since the control loops are closed through the CAN filed bus, special attention has been paid to priority CAN-identifiers. In this sense, a study of task priorities for each node has been carried out, where digital inputs provided by the laser is the most priority task, followed by the speed control loop and steering wheel control loop. A lower priority identifier has been assigned to the A/D accelerator pedal and to the A/D break pedal, because the information provided is only useful for data monitoring. Finally, the lowest priority task is given to the IR and US sensors, due to the uncertainty provided by those sensors.

An embedded computer is used as CAN field bus supervisor. A GPS is connected via RS232 to this computer for vehicle global position, providing the latitude and longitude.

Fig. 2 shows the hardware architecture proposed for this vehicle.

There are no Real-Time characteristics required for the embedded computer’s OS, because the loops are close through the field bus. In this sense, the Windows98 O.S. have been used.

A client-server application running on the on-board computer communicates wireless with a server application, which controls the whole convoying. It is interesting to remark, architecture’s modularity allows to implement new nodes and they can be easily reusable for other mobile robots purposes. Furthermore, US and IR sensors used with vehicle are similar to those ones used in the industrial forklift.

Fig. 2. Architecture for an Electric Car
2.3 Architecture for Robuter II

Robuter II is a commercial robot mainly used in research applications [6]. This robot is supplied with two DC motors driving the wheels (differential configuration), with two incremental encoders coupled to them. It has a ring of 24 US sensors connected to a CAN field bus for obstacle detection and avoidance. A vision system [3] has been installed on a pallet and tilt platform in order to allow teleoperation applications as well as artificial vision processing. The original control system of the Robuter II has been substituted by two DC servo-amplifiers to control DC motor’s speed based on the incremental encoders information and a RT CPU platform from NI where three modules are connected to the RT CPU through the PXI bus. These modules that have also RT characteristics are an analogue I/O card for generating speed’s references to the DC servo-amplifier, a counter/timer card for connecting the encoders and a CAN interface card in order to communicate with the US sensors. The RT CPU have been programmed with Labview6.1i (with RT add-on). The RT OS is completely transparent to the user, because any application program can be created in a desktop computer and then downloaded into the target.

The robot position control loop is closed though this RT platform, where encoders information and US sensors are used for robot position estimation. A web-cam connected to a computer through USB have been installed on Robuter II for vision-based mobile robots applications [3], where data monitoring is feasible due to the wireless connection to a desktop computer. Fig. 3 shows the implemented architecture for Robuter II.

The main characteristic of this architecture is the use of the RT CPU, with RT cards.

2.4 Architecture for PCBot II

PCBot II is a small robot developed by the “Mobile Robotics” research group. This robot is composed by: a mechanical platform with two driving-wheels (differential configuration); two DC motors driving the wheels with two incremental encoders coupled to the shaft of each motor; and two DC driver-amplifiers for PWM signal generation. A ring of 4 IR sensor nodes has been distributed around the robot, where each node handles with 6 IR sensors, measuring different 3 zones, where 2 IR sensors are used for each zone. The speed control loop can be closed with a combined PWM & Encoder CAN-node, able to read incremental encoder measurements and generate the appropriate PWM signal for the DC motor. This node receives the desired wheel’s speed reference through the CAN field bus and transmits the wheel’s speed through the same bus. All information provided by the CAN-nodes is monitored with a PCBisquit running with RTLinux O.S. A black-board data structure manages with the CAN messages. In fact, an input black-board is used to store the transmitted sensors measurements while an output black-board is used to transmit the speed’s references. The on-board computer communicates wireless with a desktop computer, where a joystick can be connected for teleoperating applications, as shown in Fig. 4. It is also possible to extend the proposed hardware architecture due to its modularity, providing the mobile robot with other type of sensors, such as laser, US sensors or a web-cam, etc.
It can be seen that the main purpose of this architecture is to be as simple as possible and to be open in order to extend CAN node in further applications. RTLinux as O.S. provides robustness to the system due to its RT characteristics. Furthermore, due to the small size of this platform, it is used is most of the cases as a test bed for all kind of algorithms.

3 Conclusions
The problem of hardware architecture design for mobile robots has been studied in this paper. In particular, four different architectures have been proposed based on several implementations, analysis and validations.

Arobust platform based on a PLC has been introduced as the hardware architecture for an industrial forklift. This robot uses range sensors like laser sensors, US and IR for environment reckoning. Using these sensors, the CAN-nodes described in subsection 2.1 provide the desired hardware architecture characteristics: modularity, portability and solution cost.

An interesting hardware architecture for an electric car based on CAN-nodes is shown in subsection 2.2, where the control loops are even closed through the field bus, using an embedded computer as bus supervisor.

Two RT platforms are described for Robuter II and PCBot II in subsections 2.3 and 2.4 respectively. Where Robuter II uses a RT CPU with different RT cards, obtaining the desired modularity and RT characteristics, meanwhile PCBot II has also RT characteristics due to a PCBisquit with RTLinux as O.S., where it is interesting to remark that a blackboard data structure is used for managing with CAN messages.

Acknowledgment
We thank A. Herráez and S. Fuster for assisting in the implementation of architectures described in Section II.

References: