Microprocessors and Microsystems 25 (2001) 159–166

MICROPROCESSORS AND MICROSYSTEMS

www.elsevier.nl/locate/micpro

Distributed architecture for control and path planning of autonomous vehicles

J.L. Lázaro*, J.C. García, M. Mazo, A. Gardel, P. Martín, I. Fernández, M. Marrón

Electronics Department, University of Alcalá, EP Campus Universitario, 28871, Alcalá de Henares, Spain

Received 12 May 1999; revised 3 April 2000; accepted 23 February 2001

Abstract

This paper presents a new distributed architecture designed and built in order to execute, among other functions, on-board movement control and path planning of a mobile robot by using Digital Signal Processors (DSP) and NeuronChip® Microcontrollers. Different functions have been asigned to independent processor modules interconected through a Serial Communications Link that uses the LonWorks® Protocol. Each module acts as a 'node' in a data network, were commands and control data flow as short messages increasing the system flexibility. High level tasks can be developed by DSP nodes, while low level and communication tasks are carried out by the NeuronChip. An application to Path-planning with Obstacle Avoidance using this architecture is described in detail: a DSP node, using appropriate sensory information, creates a model of the environment detecting obstacles, then it computes a path to follow and executes a full tracking algorithm, including position estimation and velocities commands generation; two NeuronChip nodes are responsible of motor control and state feedback, accepting velocities commands, computing a digital closed loop algorithm and giving back sensed velocities and encoders sampling. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: DSP; NeuronChip; Movement control; Path planning

1. Introduction

Mobile robots guidance general schema consists of finding a path from a beginning point to a destination through intermediate points placed in free space [3,13]. Some algorithms establish feasible paths to reach the destination [4,5,11], but require dynamic updates on the obstacle map to rule out errors in reaching the goal. The guidance of an autonomous mobile robot is achieved step by step reaching intermediate goals extracted from the information received by the sensorial feedback loop [6].

To correct guide the mobile, besides detecting obstacles and calculate points to trace the path to follow, it is necessary to generate the path between current position and destination one. Papers dealing with control of the robot movement in real time are Refs. [1,7,10,12].

In order to get a Real Time Control of robot movements, that implies computing in short time slots several strategies and algorithms as the ones referenced previously, it is necessary to have a hardware architecture that could address this problem: just in one sampling period to complete the whole control loop, beginning with the sensory information processing and ending with the driving signals provided to motors.

The mobile robots group of the Electronics Department of the University of Alcalá has been working for years in solving this problem. In Ref. [2]it describes a microcontroller based system with a serial communication bus, that accomplishes the low level control of two motor wheels; commanding is done using messages sent from the path planning unit, and then exciting the motor drivers. In Ref. [9]it shows an architecture to manage the autonomous mobile guidance, based on microprocessors and compatible computers, and Ref. [8]describes an architecture for executing several control strategies in the guidance of a wheelchair. Also, there are some other systems based on 80 × 86 processors including path planning and trajectory control algorithms applied to autonomous vehicles [6].

The work presented here combine the experience of all such systems and propose a new architecture, based on microprocessors and microcontrollers, to control the movement of a robot. It has a layered and modular structure, dividing all the functions needed to get the high and low level control of a mobile vehicle among different processors with different performances, all of them linked with a common serial communications bus and exchanging orders and information as messages.

---

* Corresponding author.
E-mail address: lazaro@depeca.alcala.es (J.L. Lázaro).

PII: S0141-9331(01)00108-9

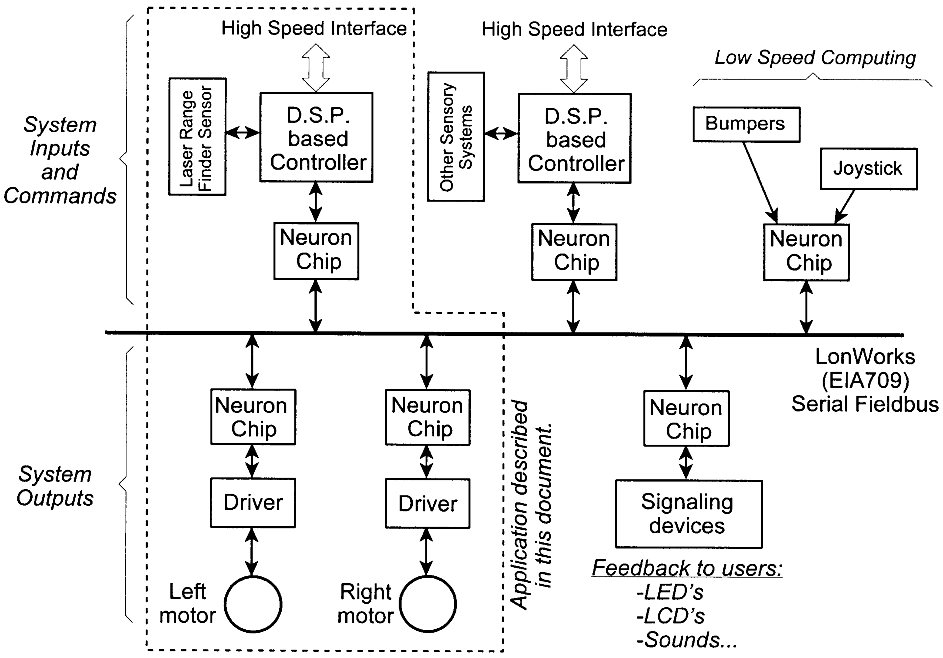

Fig. 1. General blocks diagram of system architecture.

2. Distributed architecture for mobile robots control

The distributed architecture used to control an autonomous mobile robot is shown in Fig. 1. Several modules have been included in this diagram, although in its simplest version only a few of them are required. More or less functionality can be given to the system by simply adding or removing modules.

In a closer look at Fig. 1 the following elements of the hardware architecture can be observed.

- The common link among all the modules is a Serial Fieldbus, using the industry standard LonWorks Protocol [14], recently adopted in its lower layers by the Electronics Industries Alliance with the number EIA-709. It provides transfer bit rates till 1.25 Mbit/s with several transmission media, including wireless links. This bus is an appropriate solution when short messages must be sent in known and short time slots.
- Connections to this bus are easier using the NeuronChip microcontroller, developed by the designers of the LonWorks Protocol. Although now it is an open standard and it can be built over other processors, the NeuronChip itself is both a controller and a communications device because it can execute some low level control algorithms. So, it has been chosen for those nodes with low speed computing needs as a Joystick input, bumpers detection and even as a motor controller; this specific task will be explained later.
- High speed and volume processing are carried out by Digital Signal Processors based Modules. Those nodes can process information given by high data rates sensory system, like vision or some complex ultrasonic devices and can compute the control and path planning algorithms. Also they have been designed with a high speed data interface to provide an upgrading port to connect higher level processors, if needed.

The application described in this document will need only three modules to be carried out, they have been referred to, in Fig. 1, as the ones inside the dotted box: two motor nodes, one for each motor of the vehicle (left and right); and one processor node, using a DSP node connected to a Laser Range Finder sensor.

2.1. Low level control modules

The developed system was designed with the idea of the NeuronChip itself carrying out the complete closed loop control of the motors, eliminating the need for any additional controller. Two controller boards were mounted over a platform with two driving wheels and two castor wheels. The desired speed were sent as a command message through the LonWork bus to NeuronChips acting also like motor controllers. Variables controlled were the angular speed of each driving wheel to achieve the linear speed and the turning angle of the mobile robot.

The robot is moved by a pair of DCmotors controlled by two Hbridges. The Hbridge transistors are driven by specific drivers that receive a PWM signal, with a duty cycle that indicates the motor speed [13]. A closed loop system was constructed to control the speed of the robot's wheels. This loop is closed through a system of encoders that determine the position of the motor at all times.

Experiments have proven that the NeuronChip has all the necessary support to carry out all the control tasks developed in the node described.

- It obtains the movement commands via LonWork bus messages.
- It executes a control algorithm loaded into the chip itself and written in a high level language, the 'Neuron C', similar to ANSI-C, but with quite a different functionality.
- It can gauge the encoder signals to obtain the real speed of the motor, a task carried out through one of its 32 Input/Output devices, not all of them available at the same time.
- It can generate a PWM signal by simply indicating to another one of its I/O devices the desired working cycle. This signal goes directly to the driver of the transistor bridge.
- It provides status feedback to higher level nodes, sending a message with a customized data structure. In this application it sends actual and accumulated encoder pulses, converted speed and timing information.

Modifying the application software is very simple, thanks to multiple functions included in the NeuronChip's firmware. Although execution times are rather slow, the great time constant (over 40 ms) and a time delay of 5–10 ms found in typical traction motors means that a simple digital PID algorithm can be executed without any great problem. No additional circuit is therefore necessary to carry out all the low level control of the robot's wheels, since each node acts as an intelligent component of the system. It should also be noted that each node includes all communications support.

There are two speed control loops (as shown in Fig. 2) to establish the desired trajectory.

- The first control loop operates directly over the angular speed, of the right and left wheels from the information obtained from the optical encoders placed in the axes of the motors.
- The second control loop operates over linear and angular speeds (V and Ω) of the robot itself. This allows simple trajectories to establish (lines and curves), where the curve radius comes from the ratio between the two angular speeds, of the driving wheels.

In the current architecture the robot is guided by means of commands coming from another node, higher in the control hierarchy, that computes transformations among the different kinds of speeds. So, velocities commands and data messages are referred to the angular speeds, sensed and controlled, coming directly from the right and left DCmotor nodes.

Sampling rates were fixed to 10 ms in the first loop and 20 ms in the second one. Low level nodes execute their algorithms in less than 7 ms, while high level nodes will need (solving the proposed task) less than 15 ms. The maximum message transaction time is 10 ms (from application to application software levels) using a simple twisted pair cable at a speed of 1.25 Mbit/s. This data shows that computing, sampling and transfer rates are enough for this application.

2.2. High level control modules.

Not all the functions in mobile robotics can be assigned to low speed processors; that is why it has developed a module (a single board) able to carry out higher level functions. This is also important for its use in the industry and a possible future commercialisation in the robots market. Fig. 3 shows a schema with the general blocks diagram and the functions performed by each one in the application that will be explained later.

The core of the High Level Processor Module is a TMS320C31 DSP, that is responsible of board management and execution of the algorithms; its principal characteristics are Harvard architecture and floating point arithmetic. The main processor is helped by a XC3064 LCA that accomplishes some auxiliary functions. EPROM memory stores the boot-up code program and data variables changed during its execution uses the RAM memory. Interfacing to the Serial Bus is made through a NeuronChip, this time it is

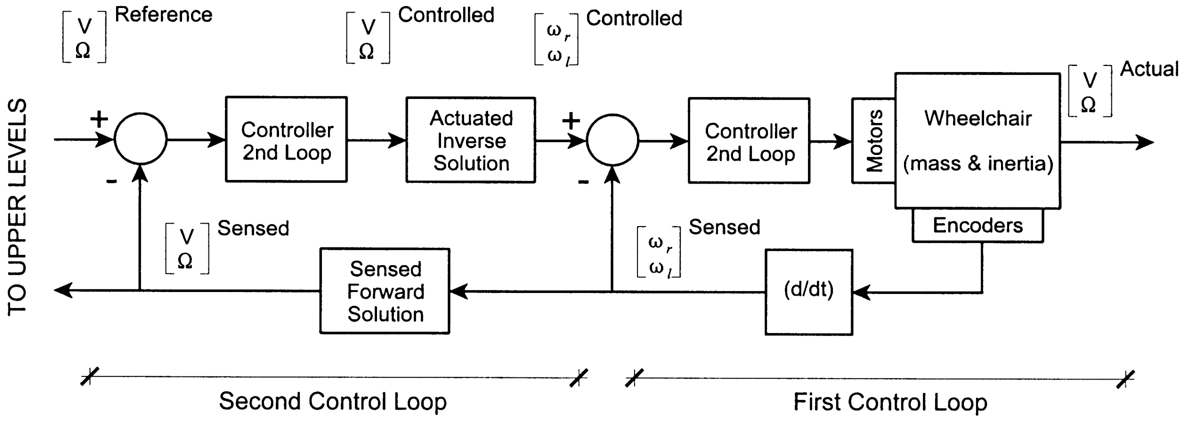

Fig. 2. Velocities double control loop in the mobile robot.

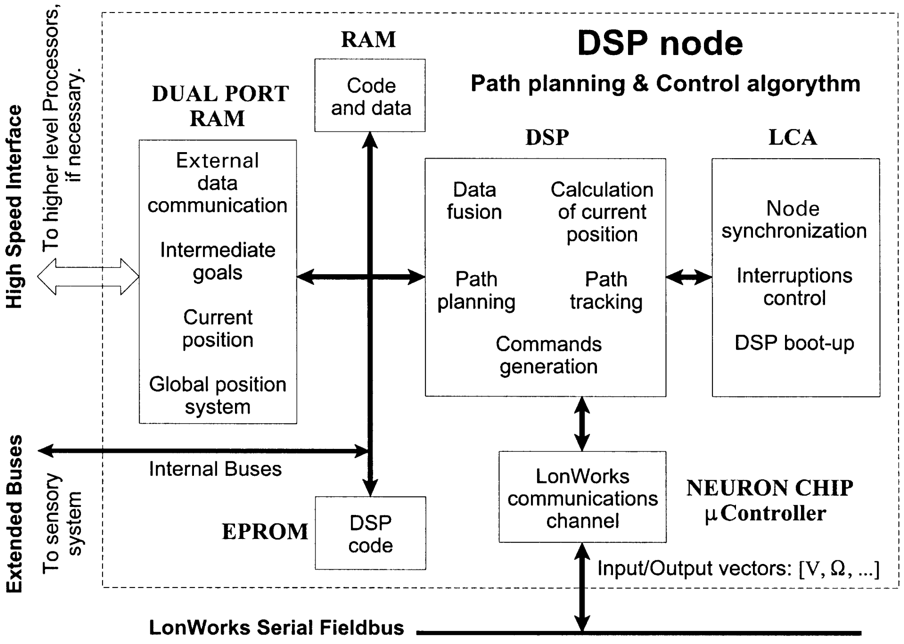

Fig. 3. Blocks diagram of the DSP board. Functions carried out by each block in this application is also shown, including path planning and tracking algorithms.

used only as a communications co-processor. The sensory information is captured directly by DSP by using an appropriate auxiliary circuitry, external to the module.

In order to provide an easy way of high speed interfacing to DSP, a Dual Port Memory has been added to the module. This kind of memory interface allows a flexible communication with even higher level processors or for debugging procedures, using a standard PC for instance. In the dual port memory, any variable that should be shared with the rest of the system can be stored. Arbitration in access to memory is made by using interruptions.

3. An application: path planning with obstacle avoidance

Here an application of this architecture to a Real Time navigation problem is described. A mobile platform with two driver motors and a Laser Range Finder must reach an objective point without a previous knowledge of the environment and avoiding any obstacle presented in the way. As it was shown in Fig. 1, only three modules are enough for this task: the high level processor, carrying out the sensory processing and the path planning, and two low level motion controllers. Using the LonWorks bus as communication link, low level nodes receive linear and angular speed commands from the high level node and they return status information like velocities measures and accumulated encoder pulses.

A blocks diagram of all the processes included in the high level node is shown in Fig. 4. There can be distinguished three parts: path planning with an intermediate goals selection stage; a tracking control loop and a communications block with low level nodes.

Sensory information comes from a Laser Range Finder also developed by the author [6]. Range data are made up of hundreds of 2D relative coordinate points of targets in the field of view of the sensor camera, already filtered and grouped; this sensor has been designed as a kind of intelligence that can simplify pure control processes by giving reliable data to the main processor. Further details about the range finder is out of the scope of this document.

3.1. Path planning

Given the environment data coming from the sensory system, the obstacles map is updated considering the current position of the robot as the origin of the reference system. Further data processing simplifies obstacles shape and try to find a feasible path between initial and destination points. First of all, the path planning process asks for a direct path to the goal but if intersections with obstacles exist, an intermediate point is selected, generating a new path that is returned to the tracking control loop. In a negative case, it will return by the direct path. In any case, paths are traced using spline models, because it provides soft curvatures without sharp orientation changes.

Path finding is constrained by the shortest distance between initial and destination points. In case of obstacles in the way, possible intermediate goals are evaluated taking into account the global distance in each possible path among obstacles, and the shortest one is chosen.

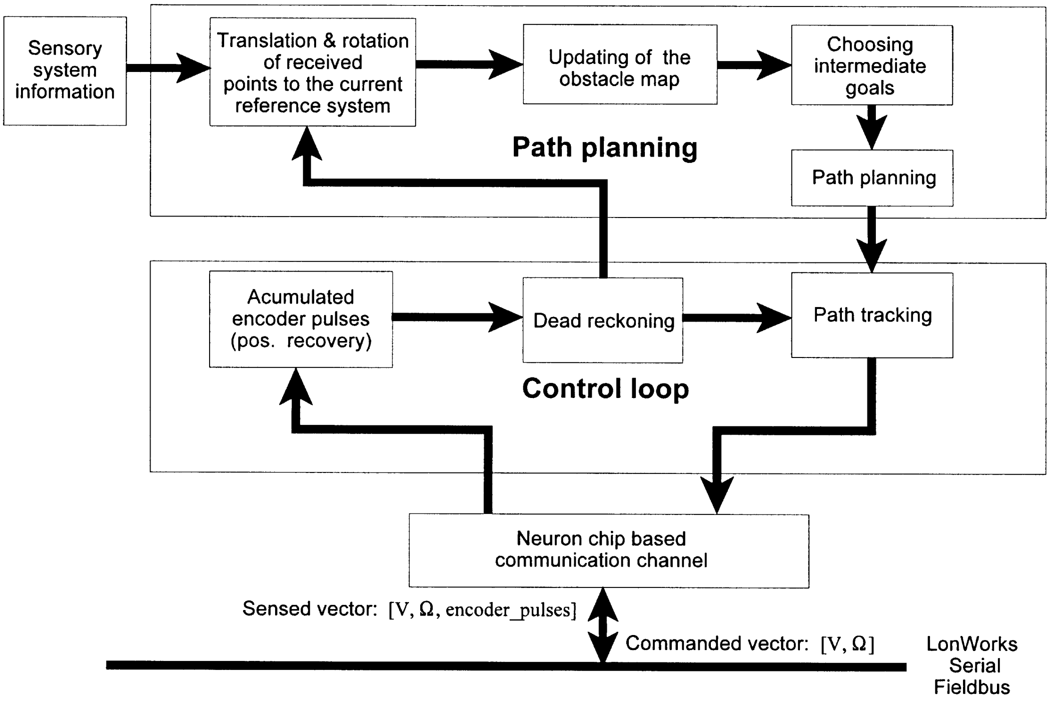

Fig. 4. Function diagram of control algorithms executed by the high level node.

3.2. Control loop and path tracking

The control loop for path tracking requests the accumulated pulses of wheels encoders, and through a dead reckoning system, calculates current position and orientation of the mobile, this data are introduced into the path planner as feedback signals. Path tracking algorithms are also responsible for generating movement commands in order to follow precisely the ideal trajectory given by the Path planner, so that the robot will be shifted away, as far as possible, from the curve defined by the computed path.

Fig. 5 shows an equivalent feedback loop diagram of the path tracking algorithm. Once the actual and desired position (P) and orientation (O) of the robot is known, an ideal corrected path, using the spline model, is generated; then, this ideal path is compared with the real one and errors will be processed by a PID controller in order to give correction values to angular speed. Linear speed, as shown in Fig. 5, is not affected by the path tracking algorithm but by the higher level path planning block. Final velocities vector, with the desired linear and angular speeds, is sent to low level nodes. Loop is closed by the sensed vector, coming from low level nodes, that includes all the information needed by all the high level processor tasks.

All the algorithms and tasks described here can be easily identified also in the hardware blocks diagram in Fig. 3. Most of the processing is carried out by DSP, while LCA and NeuronChip microcontroller help it in some decentralized processes. Configuration and monitoring data were located in Dual Port Ram, most of them for debugging and testing directly from the parallel port of a standard PC.

4. Results

Fig. 6 presents a general view of the experimental

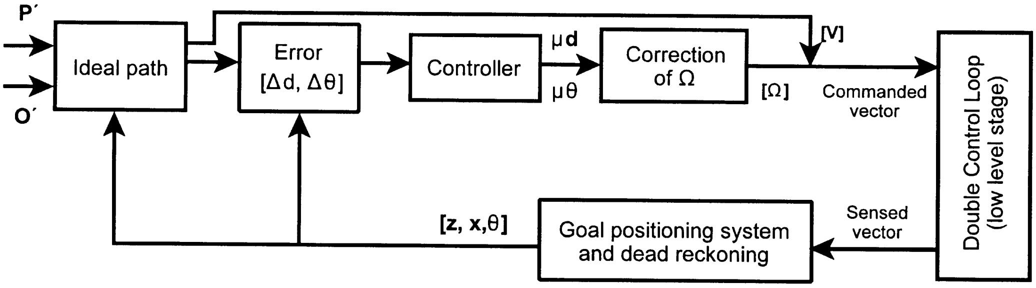

Fig. 5. Path tracking control loop.

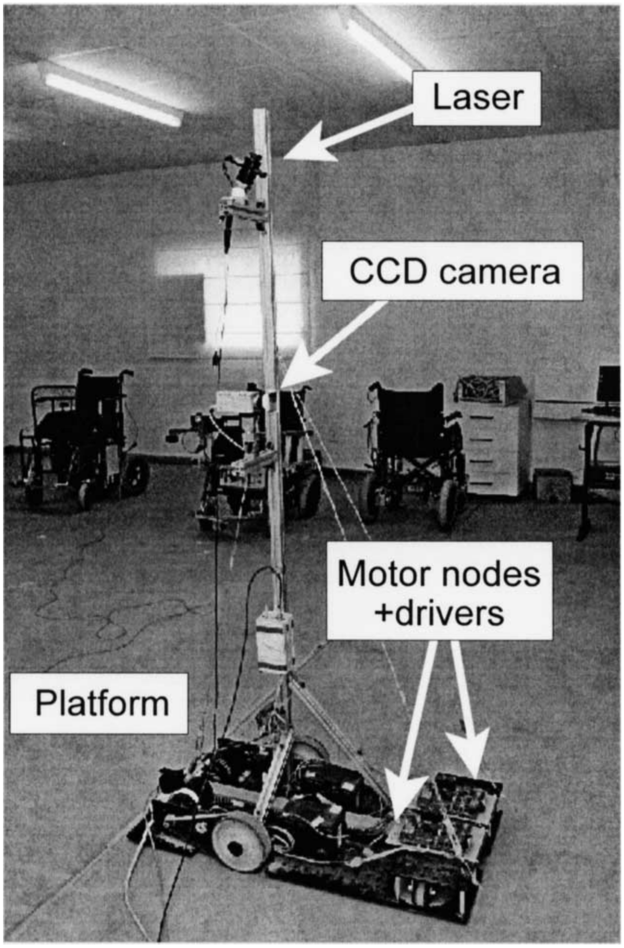

Fig. 6. General view of the experimental platform used in tests.

Table 1
Results with the evaluation of the 'dead -reckoning'

	Mean error	Max. error
Error X	3.2 cm	5 cm
Error Z	1.8 cm	5.9 cm
Error θ	0.3°	< 5°

Table 2
Profiling times in path planning and in execution of the main control loop.

Process	Average time employed (ms)
Get sensorial data + Change reference system + Update obstacles map + Calculate intermediate goal	11
Compute current position	< 1
Command generation	3

platform used in final tests, it is 90 cm long and 45 cm wide. The Laser Range Finder on board is a first prototype that uses a conventional CCD camera connected to a PC with a PCI Frame Grabber. A very reduced version, using miniature digital CMOS cameras and a single DSP, is under development. Nevertheless the flexibility of the Serial Bus used is well demonstrated because the DSP node was physically connected to the PC at a distance of several meters away from the mobile robot and, although a wireless connection was possible, a cheaper twisted pair cable was enough for testing.

Translating design to the real world, sampling rates were fixed to 10 ms in low level control loop (motor nodes) and 20 ms in the high level one (DSP node), taking in account the dynamic characteristics of the mobile robot used.

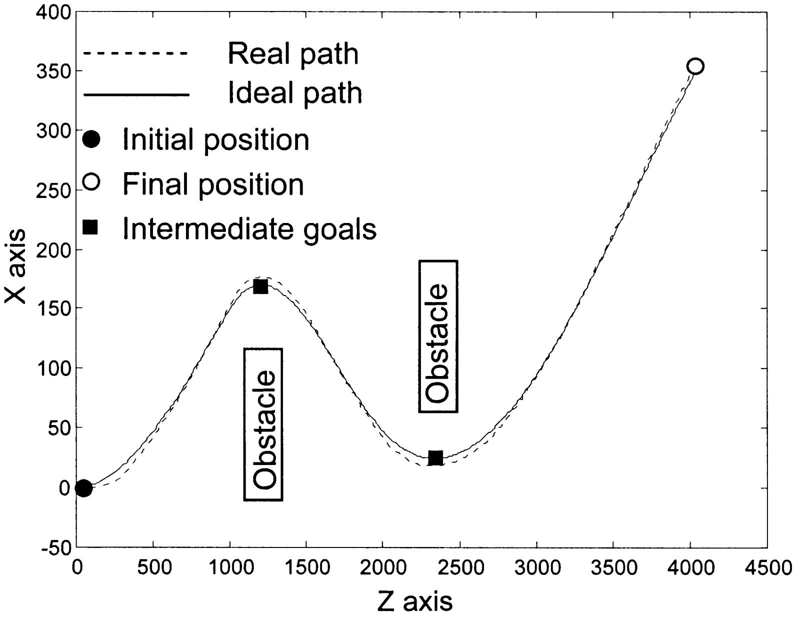

Fig. 7. Ideal and real paths, planned with spline based algorithms.

Table 1 shows the mean and maximum errors measured in the position and orientation estimation by the odometric system, following different paths. It can be observed that results obtained are precise enough in order to suppose the robot state estimation as valid. There is global data coming from more than 20 trials with trajectories like the one shown in Fig. 7 with several left and right turnings, and also including multiple circular paths.

Table 2 shows a measure of the mean time spent in path planning and in the execution of tracking control loop in the same trials set. Within the tracking control loop, one must emphasise the time of receiving the data from the actual sensory system.

Low level nodes execute their algorithms, a simple digital PID, in less than 7 ms, while high level nodes will need, to solve the proposed tasks, less than 15 ms, as can be seen in Table 1. Maximum message transaction time is 10 ms (from application to application software levels) using a simple twisted pair cable at a speed of 1.25 Mbit/s, but low level nodes feedback their data once in each 20 ms high level sampling period. This data shows that computing, sampling and transfer rates are enough for this application.

Fig. 7 also shows an example of empirical results obtained in tests. It shows two traces: one is the ideal path (commanded) and the other one is the actual path followed by the mobile platform. Dimensions have been scaled in millimeters. Obstacles were placed at distances between 1 and 1.5 m, more or less the same size as the mobile robot. Maximum linear speed in tests was limited to 1 m/s, due to several constraints in image processing algorithms in the Laser Range Finder sensor prototype. Real time path planning and tracking algorithms were programmed in standard C, because of its portability to other systems, likeTexas Instrument's C6x DSP for instance.

5. Conclusions

This work has presented a new solution to the build up a distributed control system able to control mobile robots at a very low cost and with reasonable performances. It is based on several modules, with different degrees of computing capabilities, each of them appropriate to carry on high and low level functions in the system. More or less functions can be added to or removed from the mobile robot simply by adding, removing or changing the application code of system modules. Modules have been designed with control specific microcontrollers and DSP's, making this architecture interesting because its low cost and high response speed to real time events.

A strong element of the architecture is the Serial Fieldbus, an industry standard, that allows commanding and state feedback among modules in a known and limited time slot. Using messages also increases system flexibility, because changes in design specifications can be done only by changes in software.

The power of the architecture shown is increased by using intelligent sensory systems, that could provide the high level processors on-board the robot with enough, and pre-processed, information about the environment. An example using a Laser Range Finder has been tested successfully. With only a raw pre-filtered 2D distances to obstacles data, main DSP board was able to carry on all the tasks needed for path planning and tracking avoiding obstacles with reasonable precision and high sampling rates and a low processing time.

Acknowledgements

Prototypes, hardware and tests presented in this work have been possible to finish as part of the approved project, subsidised by the University of Alcalá with reference E0008, that it is under development in its Electronics Department.

References

[1] R.A. Brooks, A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation RA-2 (1986) 14–23.

[2] J.C. García, M. Marrón, J.A. García, J. Ureña, J.L. Lázaro, F.J. Rodríguez, M. Mazo, M.S. Escudero, An autonomous wheelchair with a LonWorks network based distributed control system. Int. Conf. On Field and Service Robots FSR '97, Canberra, 1997b, pp. 420–425..

[3] S.K. Kambhampati, L.S. Davis, Multiresolution path planning for mobile robots, IEEE Journal of Robotics and Automation RA-2 (1986) 135–145.

[4] E. Koch, C. Teh, G. Hillel, A. Meystel, C. Isik, Simulation of path planning for a system with vision and map updating. Proceedings of the IEEE International Conference on Robotic Automation, St Louis, MO, 1985.

[5] B.H. Krogh, D. Feng, Dynamic generation of subgoal for autonomous mobile robots using local feedback information, IEEE Transc. on Automatic Control 34 (5) (1989) 483–493.

[6] J.L. Lázaro, Modelado de entornos mediante infrarrojos. Aplicación al guiado de robots móviles. Thesis Doctoral. E.P. Universidad de Alcala, 1998.

[7] J.L. Lázaro, A. Gardel, M. Mazo, C. Mataix, Guidance of autonomous vehicles by means of structured light. IFAC Seminar on Intelligent Components for Vehicles, Sevilla, 1998.

[8] P. Martín, M. Mazo, I. Fernández, J.L. Lázaro, F.J. Rodríguez, A. Gardel, Multifunctional and autonomous, high performance architecture: application to a wheelchair for disable people that integrated different control and guidance strategies. Microprocessors and Microsystems,1999.

[9] M. Mazo, F.J. Rodríguez, M. Sotelo, J. Ureña, J.C. García, J.L. Lázaro, F. Espinosa, Automation of an industrial fork lift truck, guided by artificial vision, ultrasonic and infrared sensors in open environments. International Conference on Field and Service Robots, FSR '97, Canberra, 1997, pp.192–199.

[10] J.J. Nitao, A.M. Parodi, A real-time reflexive pilot for an autonomous land vehicle, IEEE Const. Syst. Mag. 6 (1986) 10–14.

[11] B.J. Oommen, Robots navigation in unknown terrains using learned visibility graphics Part I: disjoint convex obstacle case, IEEE Journal of Robotics and Automation RA-3 (1987) 672–681.

[12] D.W. Payton, An architecture for reflexive autonomous vehicle control. Proceedings of the IEEE International Conference on Robotic Automation, San Francisco, CA, 1986, pp. 1838–1845.

[13] C.E. Thorpe, Path relaxation: path planning for a mobile robots. Proceedings of the AAAI Conference, 1994.

[14] hhttp://www/echelon.com

J. L. Lázaro received the Electronic Engineering and Telecommunication Engineering degrees from the Polytechnical University of Madrid (Spain) in 1985 and 1992, respectively; and his PhD degree in Telecommunication from the University of Alcalá (Spain) in 1998. Since 1986 he has been a lecturer at the Electronic Department of the University of Alcalá. During this period he has collaborated in several research projects in the areas of electronics and multi sensorial systems for mobile robots (infrared, artificial vision) and integration applied to the mobile robots

J. C. García obtained Electronic Engineering and Telecommunication Engineering degrees from the Polytechnic University of Madrid, Spain in 1982 and 1992, respectively. He has been a lecturer in the Electronics Department of the University of Alcalá since 1984. His areas of research are multisensor (ultrasonic, infrared and artificial vision) integration applied to mobile robots, intelligent wheelchairs for physically disabled people and electronic control systems applied to railway environments. He is currently working for his PhD in these subjects.

Manuel Mazo received the Electronic Engineering degree in 1976, the Telecommunications Engineering degree in 1982, and the Doctor of Telecommunications Engineering degree in 1988, all from the Polytechnic University of Madrid (Spain). He is currently a Professor in the Department of Electronics at the University of Alcalá. His areas of research are multisensor (ultrasonic, infrared and artificial vision) integration applied to mobile robots (and wheelchair for physically disabled people) and electronic control systems applied to railway environments.

A. Gardel obtained an Electronic Engineering degree from the Polytechnic University of Alcala, Spain in 1995. He has been a lecturer in the Department of Electronics at the University of Alcalá since 1997. His main areas of interest include computer vision, development of real-time systems and autonomous robots. He is currently working on robot motion planning.

Pedro Martín obtained Electronic Engineering and Telecommunications Engineering degrees from the Polytechnic University of Madrid, Spain in 1990 and 1994, respectively. He has been a lecturer in the Electronics Department of the University of Alcalá since 1991. His areas of research are DSP'S, electronic design of architectures applied to wheelchairs for physically disabled people. He is currently working for his PhD.

Ignacio Fernández obtained his Electronic Engineering and Telecommunications Engineering degrees from the Polytechnic University of Madrid, Spain, in 1984 and 1996, respectively. He has been a lecture in the Electronic Department of the University of Alcalá since 1985. He is currently working for his PhD. His main research interest is in artificial vision.

Marta Marrón obtained an Electronic Engineering degree from the Polytechnic University of Alcalá, Spain in 1996. He has been a lecturer in the Department of Electronics at the University of Alcalá since 1997. His main areas of interest include development of real-time systems and integration applied to mobile robots. He is currently working for his MD in these subjects.