An autonomous robotic system

Dan-Marius Dobrea
Electronics, Telecommunications and Information Technology
“Gheorghe Asachi” Technical University
Iaşi, Romania
mdobrea@etti.tuiasi.ro

Monica-Claudia Dobrea
Electronics, Telecommunications and Information Technology
“Gheorghe Asachi” Technical University
Iaşi, Romania
qmonro@yahoo.com

Abstract— **In the last years, a large number of applications (like robotic search and rescue operations, car safety systems, robotic space missions etc.) have been required an autonomous dynamic behavior of the motion in order to achieve their final particular objectives. In what follows, the aim of this paper is to present the design and implementation of a new autonomous robotic system that, besides its self-ruling behavior, also allows for an external (radio) remote control.**

Keywords-robotic system; car safety system; autonomous dynamic behavior; neural network

I. INTRODUCTION

Nowadays, in the car safety systems, the detection of the moment when the tired driver is about to fall asleep – followed by the car countermeasures in response to the driver behavior – is considered to be of major importance. Usually, the surrounding environment may involve any number of obstacles of arbitrary shape and size and, moreover, some of them may be moving while others are fixed. Whatever, when tired the reaction time of any person is normally longer than usual and consequently, it is most likely that the driver to have not enough time to stop the car in due time.

In the last decade, beside the increasing number of motor vehicles (about 600 million passenger cars) and the old incriminated causes in the car accidents (like driver behavior [1], motor vehicle speed, driver impairments - alcohol, physical impairment, old age, sleep deprivation and drug use -, road design and, not in the last, vehicle design and maintenance), the rapid increase in cell phone use (approximately 5 billion cell phone subscriptions around the world) has been exacerbated an already worsening traffic fatality rate worldwide. Statistics reveal that distracted drivers, like those talking behind the wheel, are about four times as likely to be involved in a crash as those who are focused on driving, and drivers who are texting are more than 20 times more likely to crash than non-distracted drivers. In 2008, nearly 6,000 people were killed and more than half a million were injured in crashes involving distracted driving in the U.S. alone. Related with the driver behavior, a report from 1985, based on British and American crash data, found that driver error, intoxication and other human factors contributed wholly or partly to about 93% of the crashes.

Today, in the policies implemented by the governments, in order to get higher road safeties, besides measures like designing safer roads, implementing blackspot programs, increasing the use of public transport, etc., an increasing attention are giving to the intelligent safer vehicles. Nowadays, a rising number of cars are equipped with board computers and each day we assist to the emergence of more and more improvements in vehicle safety. Examples of such emerging technologies are the alert-systems, dedicated to the tired drivers, like: the Eye-tracking technology [2] (that alerts drivers whenever they start to drift off or feel the effects of driver fatigue, this technology is based on the eyelids movements), the Driver Alert Control, the voice systems (the voice of the driver is analyzed and whenever is necessary the system performs some safety countermeasures in response to the drive indentified state) [3] and the Lane Departure Warning system [4]. Driver Alert Control is a technology that alerts tired and unconcentrated drivers while Lane Departure Warning alerts the driver if the car crosses one of the road markings without an obvious reason.

But, as good as these systems will prove to be, it is up to the drivers reaction time that the avoidance of the obstacle being in front of the car (either this is another car rolling on the road, a pedestrian crossing a street or, simply, some fixed objects) to be successfully accomplished. Normally, when tired the reaction time is longer than usual, and, consequently, there is a high chance the collision not to be avoided.

In this context, a solution for car accidents is given by the development of the intelligent safer vehicles - namely, vehicles that incorporate in their board computer some kind of autonomous control system. This control system will have the capacity to avoid obstacles, independently of the driver command. In an imminent collision situation, the autonomous control system will automatically take the control of the car and will keep it until the commands received from the driver will not lead the car in a dangerous position. Such an intelligent, autonomous control system prototype was designed, implemented and will be presented in what follows. Our promising solution to this problem is based on a self-organized intelligent robotic system, autoRobot, endowed with the ability to plan and to execute (independently from a human operator) a collision-free motion within its environment. In order to accomplish its goal the robot uses only the local representation of the external world, obtained from its incorporated sensors system, without any other type of external control.

II. TECHNICAL ARCHITECTURE

A. *The system arhitecture*

The whole system consists of three parts: the autonomous

We authors are grateful to the Texas Instruments^{TM} Company for the donation of the CC1000PP module systems. We take the opportunity to thank again, to the same company, Texas Instruments^{TM}, for its 2009 generous donation to our Faculty of Electronics, Telecommunication and Information Technology.

robotic system, autoRobot, the remote controller system and a main base system used to build, deploy, configure and monitor the last version of the released software application.

The structure of the robotic system is presented in Fig. 1. The robotic embedded platform, autoRobot, represents the system main module running the intelligent self-organizing algorithm (in our case the neural network). The autoRobot is a radio controlled system, able to move forward, backward, right and left, based on a set of corresponding commands received from the remote controller system. The autoRobot can: (1) be remotely controlled, (2) become a completely autonomic system, when the radio contact is lost, (3) execute (independently of a human operator command) a collision-free motion within its environment in order to (4) properly avoid the obstacles, without any damages and crashes.

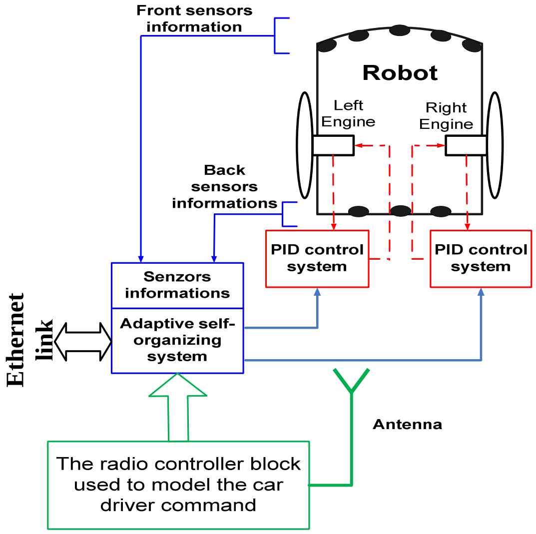

Figure 1. The robotic system technical architecture

To reach their targets or to stop without collision, the robotic systems must be endowed with perception, data processing, recognition, learning, reasoning (interpreting, decision-making) and action capacities. Mainly, because the complex environments are not predicable, pre-programming a rule-based system in order to accomplish the above-mentioned tasks is quit impossible – from here, the necessity to have an adaptive system.

The autoRobot is such an adaptive system and it has an autonomous behavior in each of the two its main working modes. In its first working mode, the robotic system behaves actually like a human being that, at the beginning of its existence, is learning to walk through a continuous learning process; this first phase corresponds to the training part of the artificial neural network (ANN). Then, based on the learned dynamics behavior, in the second working mode (that integrates the radio communication system with the ANN system), the robotic system alternates its navigation modes between goal pursuing (it obeys the radio received command) and obstacle avoidance. Exactly, in this second working mode, the autonomous behavior of the robotic system becomes dominant and takes the control anytime when the radio command sends the robot towards an obstacle and the collision is imminent (in this case the robot autonomously avoids the obstacle). In our practical implementation the radio link control models the driver commands.

B. *Hardware arhitecture*

To reach a reasonable degree of autonomy, for the autoRobot system two basic requirements are: sensing and reasoning. In order to sense the environment, the robot has 8 distance infrared (IR) measuring sensors (GP2Y0A21), Fig. 2 and Fig. 3. Five sensors are placed in front and the others three are placed in the backward positions of the robotic system.

For the reasoning requirement we used the embedded module eBox-3300A. The eBox module is almost a self-sustained system, it additionally requiring only the distance information and a power source of 5 volts. To obtain this voltage, a PTR08060W module was used, see Fig. 2.

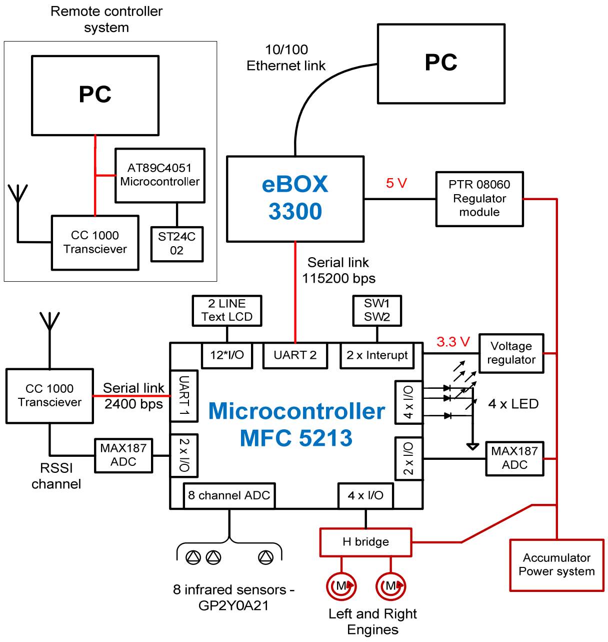

Figure 2. Details of the robotic system technical architecture

The eBox embedded system has a fast (115200 Baud rate) main serial communication backbone link with a microcontroller system. This microcontroller is one based on the 32 bits MCF5213 processor. The MCF5213 processor has several implemented functions such as: (i) whenever the eBox software sends request, it acquires the distance information from the GP2Y0A21 displacement sensors, (ii) it receives the commands through 433 MHz link from the remote control operator and, not in the last, (iii) with 2 PWM channels, based on two H-bridge structure, it controls two DC engines.

The transceiver, employed to communicate with the remote controller system, uses the CC1000PP-433 module, see Fig. 1, 2 and 3. The transceiver module is a UHF transceiver designed for very low power and for very low voltage wireless applications.

The MCF5213 was configured to interface and to communicate with the CC1000 transceiver and with the eBox, using for this two serial ports. The communications based on these serial ports were implemented using the interrupts in order to minimize the latencies and to lower the computational costs of the system.

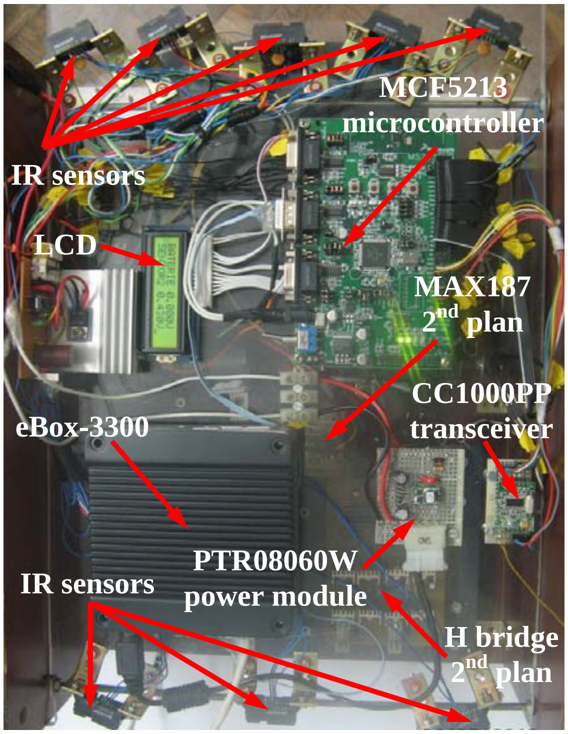

Figure 3. A picture presenting the embodiment of the autoRobot system

One of the autoRobot key concepts was to distribute the tasks between the "brain" of the system (the eBox system) and the microcontroller. Precisely, the eBox module collects the environmental (distance) information from sensors but only through MCF5213 microcontroller. In this mode, the eBox-3300A uses all the available power only for the intelligent mobile navigation module.

C. *Software architecture and algorithms*

The autoRobot software system was supported by the Windows CE 6.0 R3 operating system.

The system data flow was directly dependent on the data requirements of the adaptive ANN algorithm, used to make the autoRobot to navigate while avoiding the obstacles from its immediate vicinity. The autoRobot program was composed of 4 different threads, each of them having different priorities, see Fig. 4. In Fig. 4 the threads are represented based on their own priority; the "I/O manager thread" has the highest priority and the "primary thread" has the lowest priority.

The first thread, named "I/O manager thread", has two main functions. The first one is to send the previously obtained outputs of the ANN to the MCF5213 microcontroller; these data are the results of the forward neural network cycle – FP (forward propagation) in Fig. 4. Regarding the second function, this consists in an infrared sensor acquirement command that is sent to the MCF microcontroller. After these two operations, the thread finishes.

The second thread ("main processing thread") starts when the eBox receives the sensor information – the sequence marked with S_i in Fig. 4. In this thread, depending on the requirements of the adaptive algorithm, it may or may not include the backpropagation phase (BP in Fig. 4) and the error computation (E_n). After the thread finishes and after other 300 ms, the "I/O manager thread" is executed based on an event mechanism.

The third thread, named "maintenance thread", is executed based on a timer from 20 s to 20 s time intervals. The thread checks the accumulator status and, when the minimum voltage per cell is dropping below 1.8V, it stops the engines, and powers down the MCF5213 microcontroller, the CC1000

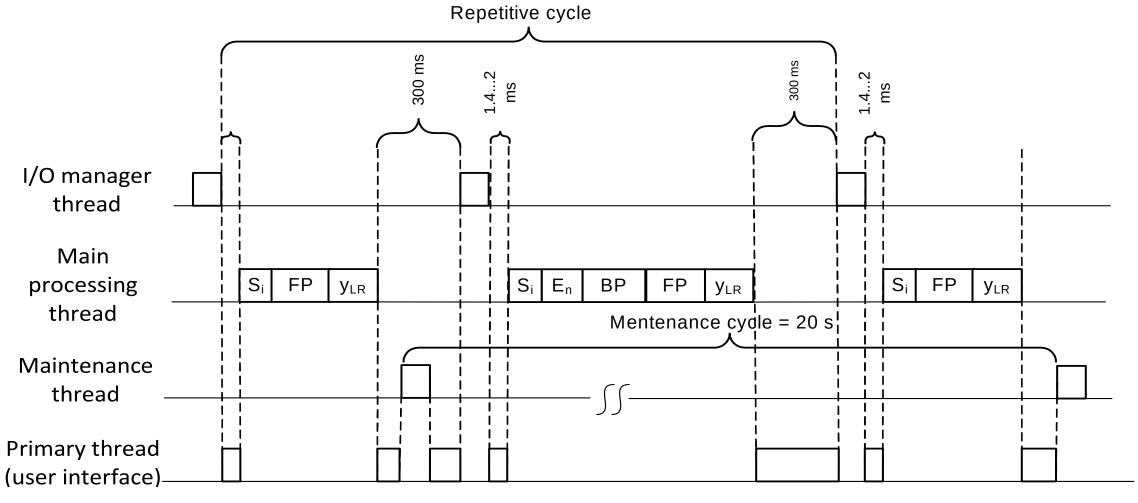

Figure 4. The thread executions and timing

transceiver and the eBox.

The “primary thread” manages only the user graphic interface. It allows to select the serial port number and the serial port communication parameters, to configure the autoRobot in one of its different working modes and, finally, to execute the software code.

The intelligent self-organizing algorithm, used to control the autoRobot, was based on a multilayer perceptron (MLP) neural network having one layer. The MLP network had a number of 8 inputs, equal to the number of the robot sensors (the MLP inputs were supplied with the normalized values obtained from the distance IR sensors). The output layer consisted in two processing elements (PEs) corresponding to the two outputs of the MLP network that supplied the commands to the engines. The inputs values, normalized in the [-1,+1] interval had the following meanings: the value -1 corresponded to the “no obstacle” case and the value +1 corresponded to “imminent collision with an obstacle” case. Each output could take values within the interval [-1, 1], with the following connotations: 1 – forward full power engine, -1 – back full power engine and 0 – stop the engine. The activation functions of the two output PEs were of tanh() function type. The network was trained using the backpropagation algorithm [5]:

$$\begin{aligned} w_{ij}[n+1] &= w_{ij}[n] - \eta \frac{\delta E}{\delta w_{ij}} = w_{ij}[n] - \eta \frac{\delta E}{\delta y_j} \frac{\delta y_j}{\delta w_{ij}} \\ &= w_{ij}[n] - \eta (d_j - y_j) s_i (1 - y_j^2) \end{aligned} \qquad (1)$$

Unlike the common applications of ANNs, where the cost function has an analytic form as in (2), depending simultaneously on the desired value, d_j, and on the corresponding output, y_j, of the network, in our particular case, we did not have a specific desired value and, consequently, the cost function was determined in an empirical mode and based on some apriori knowledge (3) and (4):

$$E = \frac{1}{2} \sum_{i=1}^{2} (d[i] - y[i])^2 \qquad (2)$$

$$E = \frac{\sum_{i=1}^{8} s_i}{8} + 1.0 \qquad (3)$$

$$w_{ij}[n+1] = w_{ij}[n] - \eta E[n] sign\left(\frac{E[n] - E[n-1]}{y_j[n]} \right) s_i (1 - y_j^2[n]) \qquad (4)$$

III. RESULTS

In the learning mode the main task performed by the autoRobot was to learn to navigate inside a delimited zone while avoiding collisions with an unknown number of obstacles randomly placed within.

The learning task was to evolve an intelligent self-organizing behavior based only on the neuronal adaptive process, see (4). This behavior was a set of stimulus-response rules encoded into the neuronal network weights. Based on these neuronal rules the autoRobot mapped the current sensors state into special engine commands that had as main objective a collisions free path. After the adaptive phase finished and the robot reached the optimal weights, several behaviors were manifested by the autoRobot.

One behavior obtained with our system consisted in avoiding an imminent collision based on the following dynamics: when the robot came closed enough to an obstacle it stopped and after that it took quickly back, making in the same time a slightly rotation. The autoRobot repeated these actions several times until it was able to have a clear path (without any kind of obstacle) in front of it. Another behavior was similar with the one of a ball that smash in a wall at an angle different of 90º. The main difference is given by the fact that the autoRobot does not touch the obstacle. With this behavior the autoRobot learned a smooth trajectory (for more details see the movies presented at the web address [6]).

In the second stage, the robot could be remotely controlled but only until the robot sensed an imminent obstacle in front of it. At this moment the neuronal network took the robot control and the obstacle was avoided based on the previously learned self organized behavior soon after that, the remotely radio control was gained back.

IV. CONCLUSIONS

In this paper we presented a remotely radio controlled autoRobot system able to self-organize and auto-evolve to a behavior that allows for sensing, reasoning and acting – all these in order to avoid all the obstacles from the immediate proximity.

As a final conclusion, the obtained results support the concept validity and opportunity of using such systems in car safety applications.

ACKNOWLEDGMENT

This work was supported by CNCSIS – UEFISCSU, project number PNII – IDEI 1552/2008.

REFERENCES

[1] H. Lum, and J. A. Reagan, “Interactive highway safety design model: accident predictive module,” Pub. Roads Mag., vol. 59, 1995, web link http://www.tfhrc.gov/pubrds/winter95/p95wi14.htm.

[2] M. Imran Khan, and A. Bin Mansoor, “Real time eyes tracking and classification for driver fatigue detection,” Img. Analy. and Recog., Lect. Not. in Comp. Sci., vol. 5112, pp. 729-738, 2008.

[3] Ford Global Technologies, “Adaptive voice control and vehicle collision warning and countermeasure system,” US Patent 7188012, March 6, 2007.

[4] [4] Parag Batavia, “Driver adaptive lane departure warning systems,” doctoral dissertation, tech. report CMU-RI-TR-99-25, Robotics Institute, Carnegie Mellon University, September, 1999.

[5] Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1985.

[6] http://www.etti.tuiasi.ro/cin/Research/WP-BioRobot 2009-2011/ProjectE.htm.