FPGA-Based Obstacle Avoidance and Line Tracking System For Autonomous Mobile Robots

1st Sorore Benabid
Electronics and Automatic Department
Electrical and Computer Engineering School (ESEO)
Paris, France
sorore.benabid@eseo.fr

2nd Loic Latour
Aeronautical and Aerospace Systems Department
Aeronautical Engineering School (IPSA)
Paris, France
loic.latour@ipsa.fr

3rd Solène Poulain
Aeronautical and Aerospace Systems Department
Aeronautical Engineering School (IPSA)
Paris, France
solene.poulain@ipsa.fr

***Abstract*—In this paper, we develop a prototype of an autonomous mobile robot using an Field Programmable Gate Array (FPGA) based control system. The robot is able to detect the obstacles that are on its way and avoid them in real-time using ultrasonic sensors. The robot can also follow a line chosen by the operator using a camera. In addition, for the robot motion two wheels and DC motors are used. The motors are driven by an H bridge and the motor speed is controlled using a Pulse Width Modulation (PWM) signal. The whole system is implemented only in Very High speed integrated circuit Hardware Description Language (VHDL) code on a Nexys 4 development board with Artix FPGA device from Xilinx that operates at 100 MHz.**

***Index Terms*—Obstacle avoidance; Line tracking; Field Programmable Gate Array (FPGA); Autonomous mobile robots.**

I. INTRODUCTION

Mobile robots are expected to perform increasingly complex tasks in various application fields, such as: space exploration, underwater research, intelligent transport systems, military, medicine, service robots, but also in all levels of education. They should be able to navigate successfully in their environment with certain level of intelligence. To reach the desired goals, mobile robots use different kinds of sensors to collect environmental information and a set of actuators for their motion and reaction. Therefore, it is necessary to use a powerful and flexible device to control and manage the set of sensors and actuators present on the mobile robot. The Field Programmable Gate Arrays (FPGA) are gaining popularity due to their reconfigurability and parallel ability [1]. The FPGAs help enable the mobile robots to achieve their target mission successfully while guarantee real-time response in a dynamic environment [2]. Moreover, the FPGAs structure is able to execute calculating algorithms with high speed and low energy cost [3] [4].

A great deal of research has focused in the last decade on the FPGA-based mobile robot navigation. Reference [5] developed a fuzzy algorithm on FPGA-based mobile robot for line tracking and obstacle avoidance purposes. Another group proposed an FPGA-based architecture for multi-robot tracking using multiple GigE Vision cameras [6]. This architecture was implemented comprising a multi-camera frame grabber and IP cores for image processing. Another team proposed a reconfigurable embedded FPGA-based vision system for Advanced Driver Assistance Systems (ADAS) applications [7]. The developed board contains a System on Chip (SOC) composed of a programmable logic that supports parallel processing, and a microprocessor suited for serial decision making. Contrary to these various achievements, in this work, we want to realize an educational platform of a simple embedded system. This platform will allow engineering students to learn how to deploy Artificial Intelligence (AI) algorithms on an FPGA board using only the VHDL code.

In this paper, we propose an autonomous robot platform for education purposes and industrial researches. This platform completes the previous work of our team [8], in which we realized an FPGA-based vision system for autonomous mobile robots. The previous system evaluated in real-time the distance between a robot and an object or obstacle in front of it. In this work, we perform other tasks namely: autonomous navigation, obstacle avoidance and line tracking. The proposed system is implemented only in VHDL code. The paper is divided into four Sections. Section II describes the architecture of the proposed system, including the robot motion, obstacle avoidance principle and line tracking algorithm. The experiments conducted to demonstrate the performances of the system are given in Section III. Finally, we conclude in Section IV.

II. ROBOT ARCHITECTURE

The proposed embedded system for a mobile robot is shown in Figure 1. We have used two wheels and DC motors with an H bridge controller for the robot motion. The desired direction and speed of the robot is determined by the control and processing unit. This unit is an FPGA development board that allows our robot to react in real-time to its exterior environment, depending on the input information given by a single camera and ultrasonic sensors. The ultrasonic sensors are used to allow the robot to avoid obstacles and the camera is used to ensure the line tracking. Obviously, we use a battery as an external power supply for the DC motors and also for the FPGA board. Furthermore, the camera requires a huge capacity of memory for the image processing without any quality losses. Thus, we chose the Nexys 4 board because

 ISBN: 978-1-61208-823-5

TABLE I
ROBOT SPECIFICATIONS

FPGA Board	*Nexys 4 DDR - Artix-7*
Board Dimension	10,9 x 12,2 cm
Logic Slices:	15 850
Block RAM	4,860 Kbits
DDR2 Memory	128MiB
Operating Frequency	100MHz
Power Supply	4.5V-5.5V
Camera	*CMOS Image sensor*
Photosensitive Matrix	640 x 480
Issuance formats (8)	RGB 565
Maximum rate:	30 fps in VGA
Pixel height	3.6μm
Output format	VGA
DC motor	*Decelerate motors ratio 1:48*
Drive voltage	12V
Ultrasonic sensor	*HC-SR04*
Operating voltage	5 V
Operating current	15 mA
Range of distance	2 cm to 400 cm
Battery	EC Technology
Capacity	22400mAh / 82.8Wh
Entry	5V / 2A
Max Output	5V / 3.4A (AUTO)
Dimension	160x80x23mm
Weight	462g

of its storage capacity and the number of available ports. The main specifications of the embedded system are listed in Table I.

We describe in the next subsections the main operations performed by the robot: robot motion, obstacle avoidance and line tracking.

A. Robot Motion

To perform properly the robot motion, the DC motor needs to receive an electrical current which intensity will directly impact the speed and rotation of the motor. We used the Pulse Width Modulation (PWM) principle to control the speed of the robot. This will make use of the FPGA board frequency to periodically supply an amount of energy to the motor and thus provide efficient speed control.

B. Obstacle Avoidance

This function was filled by three ultrasonic sensors placed at the front of the robot and oriented at different angles. Ultrasonic Sensors measure the distance (D) to the target object by measuring the time (T) between the transmitted and received wave.

$$D = \frac{T.S}{2} \quad (1)$$

where S represents the sound speed.

The avoidance obstacle can be achieved through the distance measured by the different ultrasonic sensors as described by the flowchart given in Figure 2. Once a distance is inferior to a threshold that we have set by trials, we consider there is an obstacle to avoid. Then, we compare the different distances given by the three ultrasonic sensors, if the distance to the left is greater than the right one, the robot should turn left to avoid the object on the right. Moreover, if the distance measured is inferior to the threshold of every ultrasonic sensor, the robot will do a half turn to avoid getting stuck.

C. Line Tracking

In the proposed system, as shown in Figure 3, we have chosen to track a white line on a black background reversed when filtering. Then, we chose 11 pixels aligned horizontally, from the center to both sides of the image. These 11 points are spaced by 44 pixels and will be represented by an 11-bit binary vector.

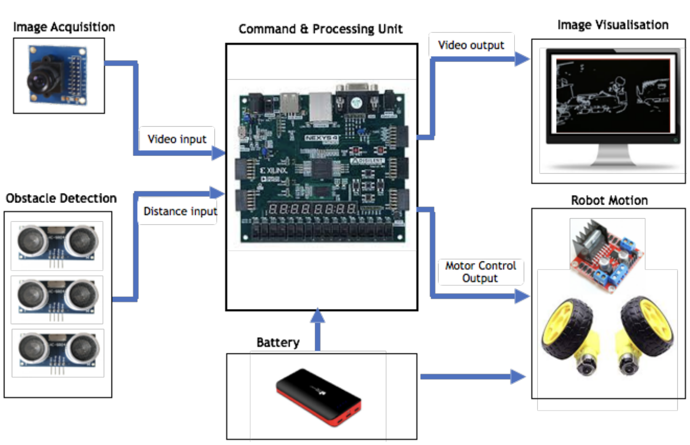

Fig. 1. Embedded system architecture for the mobile robot

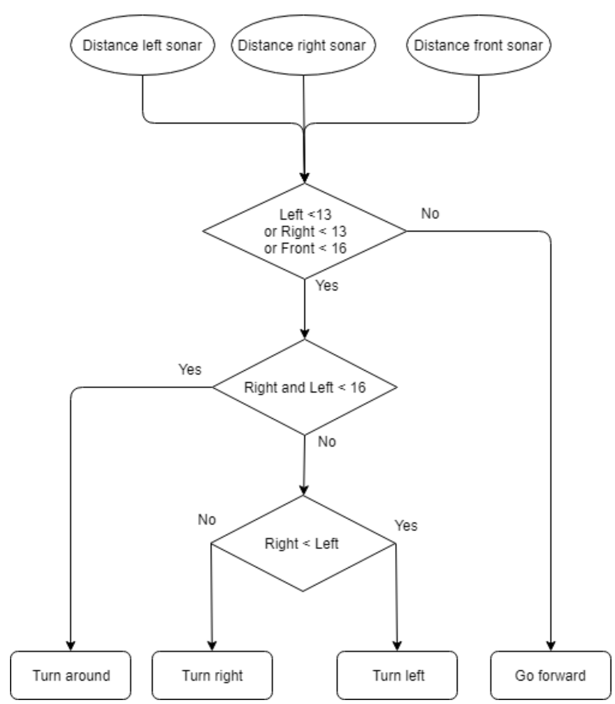

Fig. 2. Obstacle avoidance algorithm using the ultrasonic sensors

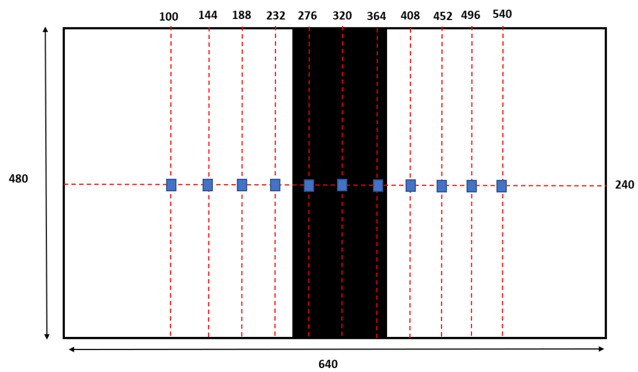

Fig. 3. Line modeling after black and white filter

To track the line, we used a single camera at the front of the robot. A black and white filter is applied to the captured image by the camera. This filter help enable to distinguish the line from the rest of the image. Each bit of the 11-bits vector are set to ”1” (white) if the pixel exceeds a threshold. Otherwise, the bits are set to ”0” (black). We adjusted the filter threshold using tests for a better result.

After the black and white filtering, the line is modeled by 3 bits at the center of the 11-bits vector that represent the 11 horizontally pixels as illustrated in Figure 3. Then, the robot can detect the line according to the value of the 3-bits modeling this line. If these 3-bits are equal to ”0” the line is detected, otherwise the white color is dominant and the line is not detected.

TABLE II
THE MOTOR CONTROL USING THE LINE MODEL

Binary vector	*Action on motor*	*Duty cycle (%)*
00111111111	Turn left	65
00011111111	Turn left	60
10001111111	Turn left	60
11000111111	Turn left	50
11100011111	Forward	50
11110001111	Forward	50
11111000111	Forward	50
11111100011	Turn right	50
11111110001	Turn right	60
11111111000	Turn right	60
11111111100	Turn right	65

Finally, we have to control the speed and the direction of the robot (left, right or center) according to the position of the ”0” in the 11-bits vector as depicted in Table II. The PWM modulation adjusts the robot speed as mentioned previously (Section II-A). However, if the robot is in a bend it loses the line, then it will stop.

III. EXPERIMENTAL RESULTS

In this Section, we describe the structure of the robot, the synthesised architecture in VHDL code and the experimental results of the achieved prototype.

A. Robot Structure Design

The structure is designed on CATIA software (Computer-Aided Three-dimensional Interactive Application). It is a multi-platform software suite for Computer-Aided Design (CAD), Computer-Aided Manufacturing (CAM), Computer-Aided Engineering (CAE), developed by the french company Dassault Systèmes [9]. The designed structure is printed by a 3D printer as shown in Figure 4. This structure allowed us to realize the robot's prototype as shown in Figure 5.

B. FPGA Architecture

The synthesised architecture is illustrated in Figure 6 and described in VHDL code. Signals from the 3 ultrasonic sensors (echo signals) are transmitted to the ”Sonar driver” block that calculated the distances of the obstacles detected by the robot. These distances are displayed by the ”7-segments” block, and are also used to determine the motion of the robot by the ”Motor driver” module. This module is used to control the

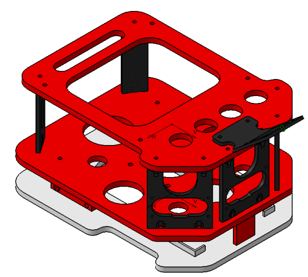

Fig. 4. Structure of the robot

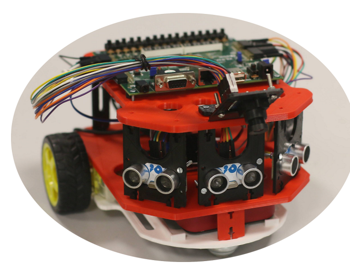

Fig. 5. The Robot

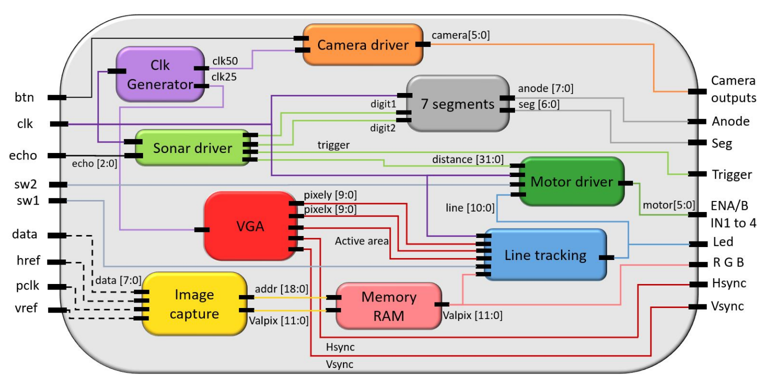

Fig. 6. FPGA-based embedded system architecture: robot motion, image processing, obstacle avoidance and line tracking

TABLE III
THE MOTOR CONTROL

IN1,IN3	*IN2,IN4*	*Result*	*ENA,ENB*
0	0	Stop	0
0	1	Rotation+	PWM
1	0	Rotation-	PWM
1	1	NOT ALLOWED	0

TABLE IV
FPGA RESOURCE USAGE OF THE PROPOSED SYSTEM

Resource	*Utilization*	*Available*	*%*
LUT	1107	63400	1,75
FF	481	126800	0,38
BRAM	104	135	77,04
IO	73	210	34,76
BUFG	5	32	15,63
MMCM	1	6	16,67

Fig. 7. Black and white filter for line tracking

DC motors via an H bridge. Using IN1, IN2, IN3 and IN4 pins we can control the direction of the rotation of the motor and thus make the robot go forward or backward (Table III). The PWM signals control the motors speed using the ENA and ENB pins. Indeed, depending on the choice of duty cycle, the intensity on these pins will be more or less important alike the speed.

The ”Camera driver” block is used to configure the camera to generate the appropriate synchronization signals with clock frequency at $50MHz$. The image of the camera is captured by the ”Image capture” block, then stored in the ”Memory RAM” block. The image is then filtered (black and white filter) and the ”Line tracking” module is used to find the position of the line as explained previously (Section II-C). The line position is then transmitted to the ”Motor driver” block that allows the robot to follow the line.

Finally, the image is displayed on the VGA screen according to the synchronization signals and the pixel position generated by the ”VGA” block. The ”CLK generator” block provides two clock signals from the FPGA operation clock $100MHz$ (Table I); $25MHz$ for the VGA screen clock and $50MHz$ for the ”Camera Driver” block.

The described architecture is synthesized and implemented in Artix-7 FPGA available on the Nexys 4 development board from Xilinx using the Vivado 2018.2 software. The proposed system does not require a processor and external RAM resources. The resource usage is summarized in Table IV.

The black and white filter for line tracking is illustrated in Figure 7. A video showing the obstacle avoidance of the proposed system can be found at: https://youtu.be/tjWPFtim8CQ. For the line tracking test, a video is available on this link: https://youtu.be/SjUswlInYgM.

IV. CONCLUSION

The emergence of FPGAs has given improvement of a real-time mobile robot navigation systems. We have proposed in this paper an FPGA-based embedded system navigation for the mobile autonomous robots. The proposed system can help the robots to navigate successfully by analysing the visual features of the surrounding environment. This system can detect and avoid the obstacles and can also track a line. This system uses a single front camera and ultrasonic sensors as sources of information on the exterior environment. It uses also two wheels and the DC motors to ensure the robot motion. The whole architecture is implemented only in VHDL parallel programming code and will be further improved and extended for other tasks such as path planning and object recognition algorithms. The achieved robot provides a good experimental platform for academic environment in collaboration with industrial partners. The engineering students will learn and test AI algorithms for mobile robots. They will also be able to propose a new AI techniques for several applications: Intelligent Transportation Systems (ITS), medical application, environmental protection, etc.

ACKNOWLEDGMENT

We would like to thank the laboratory of the autonomous aerial systems at IPSA for their daily help especially for the use of the 3D printer, different components and software tools that we have used to design and realise our prototype.

REFERENCES

[1] R. Dubey, N. Pradhan, K. M. Krishna, and S. R. Chowdhury, “Field programmable gate array (FPGA) based collision avoidance using acceleration velocity obstacles,” in 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dec 2012, pp. 2333–2338.

[2] D. Honegger, H. Oleynikova, and M. Pollefeys, “Real-time and low latency embedded computer vision hardware based on a combination of FPGA and mobile CPU,” in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sep. 2014, pp. 4930–4935.

[3] M. Samarawickrama, A. Pasqual, and R. Rodrigo, “FPGA-based compact and flexible architecture for real-time embedded vision systems,” in International Conference on Industrial and Information Systems (ICIIS), Dec 2009, pp. 337–342.

[4] J. Svab, T. Krajnik, J. Faigl, and L. Preucil, “FPGA based speeded up robust features,” in 2009 IEEE International Conference on Technologies for Practical Robot Applications, Nov 2009, pp. 35–41.

[5] S. Boroumand, A. Saboury, A. Ravari, M. Tale Masouleh, and A. Fakharian, “Path tracking and obstacle avoidance of a FPGA-based mobile robot MRTQ via fuzzy algorithm,” in 2013 13th Iranian Conference on Fuzzy Systems (IFSC), Aug 2013, pp. 1–5.

[6] A. Irwansyah, O. W. Ibraheem, J. Hagemeyer, M. Porrmann, and U. Rückert, “FPGA-based multi-robot tracking,” J. Parallel Distrib. Comput., vol. 107, 2017, pp. 146–161.

[7] G. Velez, A. Cortés, M. Nieto, I. Vélez, and O. Otaegui, “A reconfigurable embedded vision system for advanced driver assistance,” Journal of Real-Time Image Processing, vol. 10, no. 4, 2015, pp. 725–739.

[8] S. Benabid, L. Latour, S. Poulain, and M. Jaafar, “FPGA-based real-time embedded vision system for autonomous mobile robots,” in 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), 2019, pp. 1093–1096.

[9] D. Systèmes. CATIA. [Online]. Available: https://www.3ds.com/products-services/catia/

