IAV: A VHDL Methodology For FPGA Implementation

Ouarda HACHOUR and Nikos Mastorakis
Microelectronics Laboratory
Development Center of Advanced Technologies Baba Hassen Haoulh Oukil Algiers Algeria
Phone: 213 (021) 35-10-18,35-10-40,35-10-75. Fax: 213 (021) 35-10-39
E-mail: hachour_ouarda@hotmail.com
Military Inst.of Univ. Education
Hellenic Naval Academy
Hatzikyriakou 18539
Piraeus, Greece
E-mail: mastor@wseas.com

Abstract

In this paper, a flexible and compact architecture for implementing navigation approach of Intelligent Autonomous Vehicles (IAV) using a single SRAM-based Field Programmable Gate Array (FPGA) is introduced. To overcome the problem of navigation and make the robot able to achieve these tasks : to avoid obstacles, and to make ones way toward its target, a new design methodology of navigation approach based upon a VHDL description for intelligent autonomous vehicles in unknown environment is proposed. We propose a simple and regular architecture model for the navigation that allows an easier implementation . This architecture uses the Genetic Algorithms (GA) Fuzzy Logic (FL) and Expert Systems (ES) which are necessary to bring the behaviour of Intelligent Autonomous Vehicles (IAV). Also, it must make the robot able to reach its target without collision capturing the behaviour of a human expert and respond to autonomy requirements such as power and thermal. The new design methodology based upon a VHDL description of the navigation approach has the two (02) advantages : to present a real autonomy for mobiles robots, and being generic and flexible and can be changed at the user demand. To validate our approach, a VHDL description of *ES_GA_FL* is passed through a synthetic tool, Galileo for FPGA implementation.

Keywords : Intelligent Autonomous Vehicles (IAV), Genetic Algorithm (GA), Fuzzy Logic (FL), FPGA implementation VHDL , Synthesis.

I-Introduction

A robotic system capable of some degree of self-sufficiency is the primary goal of IAV. The focus is on the ability to move and on being self-sufficient . To evolve in an unknown environments for example, the recent developments in autonomy requirements, intelligent components, multi-robot system, and massively parallel computer have made the IAV very used, notably in the planetary explorations, mine industry, and highways [26,18,4,16]. This paper deals with the hardware implementation of navigation approach of an IAV in an unknown environment to give a real intelligent task and respond to autonomy requirements. The aim of this paper is to develop an IAV combining Expert Systems (ES), Genetic Algorithm (GA), and Fuzzy Logic (FL) for the IAV stationary obstacle avoidance to provide them with more autonomy and intelligence using a single SRAM based Field Programmable Gate Array (FPGA). This technology has revealed the soft-computing in order to give a high speed processing that could be provided through massively parallel implementation management and so more autonomy requirements such as power, thermal and communication management are obtained in real time. The new architecture *ES_GA_FL* can be implemented either in analogue or digital way. Digital circuits are very manufactured and are functionally identical. Analogue circuits, on the other hand, are sensitive to noise and temperature changes and inter-chips variations make manufacturing functionally identical circuits, very difficult. Nowadays, FPGA are gaining momentum in digital design . They are used for a wide range of application including rapid prototype, glue logic for microprocessor , and hard –wired simulation. Moreover , the relatively low cost, an easiness of implementation and recognition of FPGA, offer attractive features for the hardware designer in comparison with other VLSI implementation techniques. In other words, FPGA has emerged as the ultimate solution to time –to-market and risk problems because they provide instant manufacturing and very low –cost prototypes [3.6] . In this paper, we discuss the possibility to deal with this hurdle by proposing a new approach permitting the mapping of an entire *ES-GA_FL* (planning, intelligent control for obstacle avoidance) into a single Xilnx's FPGA, this new architecture presents a robust opportunity since the autonomy requirement, massively parallel computers, a real intelligent component, short time of execution, high speed processing and the objectives of navigation are obtained . The presentation is organized as follow: In section II, the proposed hybrid navigation approach combining *ES_GA_FL* is done. Section III describes a digital implementation of the *ES_GA_FL*, Section IV presents VHDL description of the *ES_GA_FL* and Synthesis and implementation results and finally, a conclusion is given in Section V.

II- The proposed hybrid navigation approach combining *ES_GA_FL*

Today, researchers have at their disposal, the required hardware, software, and sensor technologies to build IAV. More, they are also in possession of a computational tools such as FL and GA that are more effective in the design and development of IAV than the predicate logic based methods of traditional Artificial Intelligence. GA and FL are well established as useful technologies that complement each other in powerful hybrid system. The first and most advanced integration of intelligent technologies is the hybrid GA and FL. The major thrust of this type of combining GA and FL is to synthesis the capability of ES to capture expert domain knowledge in an inference –based system.

An ES is a computer program that functions, is in a narrow domain, dealing with specialised knowledge, generally possessed by human experts. ES is able to draw conclusions without seeing all possible information and capable of directing the acquisition of new information in an efficient manner [20.21].

II-2-3-1 Fuzzy Logic

As human reasoning is not based on the classical two-valued logic, this process involves fuzzy truths, fuzzy deduction rules, etc. This is the reason why FL is closer to human thinking and natural language than classical logic. The fuzzy model, treated with *ES_GA_FL* is presented in Figure 1 :

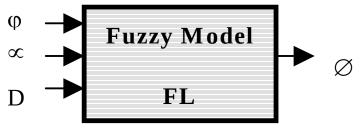

Figure 1: Fuzzy Model.

Where :

The direction φ and α are calculated by :

$$\varphi = \tan^{-1}\left(\frac{Y_i - Y_1}{X_i - X_1}\right) \qquad (1)$$

$$\alpha = \tan^{-1}\left(\frac{Y_g - Y_1}{X_g - X_1}\right) - \varphi \qquad (2)$$

Where : the $P_1(X_1, Y_1)$, $P_i(X_i, Y_i)$ and $P_g(X_g, Y_g)$ are the co-ordinate of respectively to initial point, intermediate point and visual point (we calculate point to point until the visual point becomes the target one, see Figure 2).

D: is intermediate distance between P_1 and P_i, P_i and P_g, and P_1 and P_g. The distance between P_1 and P_i for example is given by :

$$D = \sqrt{(Y_i - Y_1)^2 + (X_{i-}X_1)^2}$$

The vehicle going to a given target avoiding a static obstacle is shown in the Figure 2. At first, from a given detected position P_i (Pixel by pixel), and from a given visual position P_g, the vehicle gets absolute obstacle position P_i and P_g from current point P_1 to calculate the angles ∝ and ∝. Then, after updating its position (P_1 becomes P_g) and the target position (P_g becomes the target one) based on its Cartesian coordinates, the vehicle must avoid the obstacle to get the target position. Afterwards, the vehicle recognizes the static danger degree ∝ and safety degree ∝ between itself and the obstacle using a fuzzy reasoning and inference. Second, using ∝ and ∝ the vehicle decides the avoidance direction ∅ by a decision table written with productions rules and then avoidance direction vector . The main problem of this approach is that does not encounter several obstacles at the same time and does not take into account the obstacles sizes.

The vehicle must learn to decide ϕ using *FL* from a fuzzy linguistic formulation of human expert knowledge. This FL is trained to capture, the fuzzy linguistic formulation of this expert knowledge is used and a set of rules are then established in the fuzzy rule as it is shown in Table I and Table II.

The final decision (*defuzzification*) is achieved to give the output of fuzzy controls and to converts the fuzzy output value produced by rules. This intelligent task uses a set of rules can give a human expert knowledge.

After learning, the robot decide how to give the best direction ϕ using FL from a fuzzy linguistic formulation. The defuzzification formula is given by :

$$M = \frac{\sum (u_i * c_i)}{\sum u_i} \quad 1 \le i \le n.$$

Where :

The n is designed by the number twelve (12) (see table I and table II).

The membership labels for distance are defined as Near (N), Medium (M) , and Far (F) see Figure 3

The membership functions of directions φ and α are represented in Figure 4, where fuzzy labels are defined as left big (LB), Left Small (LS), Right Small(RS) and Right Big (RB).The membership function of direction ϕ are shown in Figure 5, where fuzzy labels are defined as : Danger on left side : Left Danger Big (LDB), Left Danger Small (LDS), Left Safety Small (LSS), and Left Safety Big (LSB) While danger on right side : Right Danger Big (RDB), Right Danger Small (RDS), Right Safety (RSS), and Right Safety Big (RSB).

IF (φ is LB and D is N) Then (ϕ is LDS)
IF (φ is LB and D is M) Then (ϕ is LSS)
IF (φ is LB and D is F) Then (ϕ is LSB)
IF (φ is LS and D is N) Then (ϕ is LDB)
IF (φ is LS and D is M) Then (ϕ is LDS)
IF (φ is LS and D is F) Then (ϕ is LSS)
IF (φ is RS and D is N) Then (ϕ is RDB)
IF (φ is RS and D is M) Then (ϕ is RDS)
IF (φ is RS and D is F) Then (ϕ is RSS)
IF (φ is RB and D is N) Then (ϕ is RDS)
IF (φ is RB and D is M) Then (ϕ is RSS)
IF (φ is RB and D is F) Then (ϕ is RSB)

Table 1 : Fuzzy Rule Base 1

IF (∝ is LB and D is N) Then (ϕ is LDS)
IF (∝ is LB and D is M) Then (ϕ is LSS)
IF (∝ is LB and D is F) Then (ϕ is LSB)
IF (∝ is LB and D is N) Then (ϕ is LDB)
IF (∝ is LS and D is M) Then (ϕ is LDS)
IF (∝ is LS and D is F) Then (ϕ is LSS)
IF (∝ is LS and D is N) Then (ϕ is RDB)
IF (∝ is LS and D is M) Then (ϕ is RDS)
IF (∝ is RB and D is F) Then (ϕ is RSS)
IF (∝ is RB and D is N) Then (ϕ is RDS)
IF (∝ is RB and D is M) Then (ϕ is RSS)
IF (∝ is RB and D is F) Then (ϕ is RSB)

Table II: Fuzzy Rule base 2

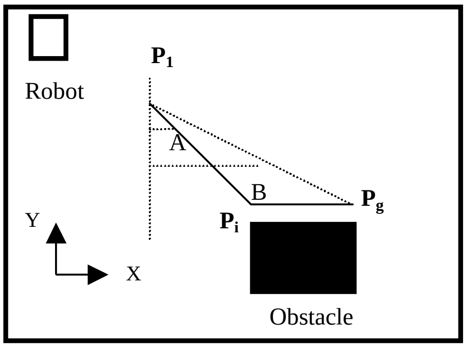

Figure 2: Vehicle obstacle avoidance mode

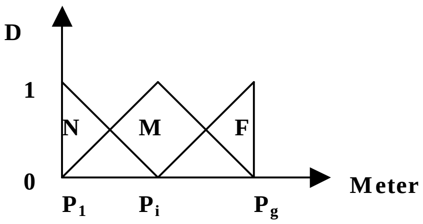

Figure 3 : Membership functions of D

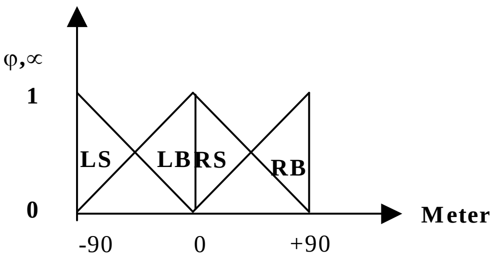

Figure 4: Membership function of directions φ and α

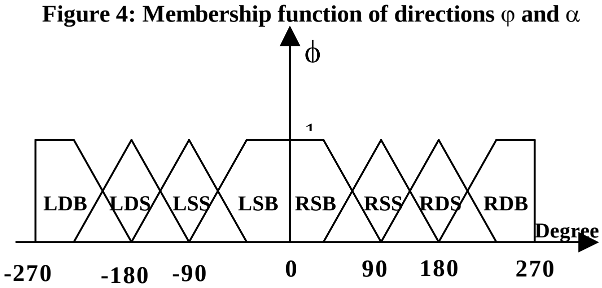

Figure 5 : Membership functions of directionϕ

II-2-3-2 Evolutionary Algorithms

The genetic algorithms which are evolutionary, have recently emerged from study of the evolution mechanisms and are searching strategies suitable for finding the globally optimal solution in a large parameter space. They are based on learning mechanism. GA's has been theoretically and empirically proven to provide robust search capabilities in complex spaces offering a valid approach to problems requiring efficient and effective searching [11]. Before the GA search starts, candidates of solution are represented as binary bit strings and are prepared. This is called a population [17]. A candidate is called a chromosome (in our case : the path is a chromosome and positions are the genes). Also, an evolution function, called fitness function, needs to be defined for a problem to be solved in order to evaluate chromosome. As fitness function, we define estimate distance for each chromosome to give an evaluation function. This evaluation is the goal of the GA search and goes as follows : two (02) chromosomes are chosen randomly from a population are mated and they go through operations like the crossover to yield better chromosomes for next generations. Because the population size is fixed, the offspring produced by genetic manipulation process are the next population to be evaluated. We must specify a stopping criterion to determine execution of the GA, we use the theorem of the combination in probability : 2^n (here n is the number of the all paths). This search mechanism is associated with mutation a mechanism of probabilistic to change a bit 0 to 1 or vice versa [8.27]. The GA treats mutation only as a secondary crucial operator after crossover. The crossover is the comparison operator, this operation takes chromosomes and swaps part of their genetic information to produce a new chromosome.

III- Digital Implementation of *ES_GA_FL*

Configurable hardware is an approach for realising optimal performance by tailoring its architecture to the characteristics of a given problem. When the characteristics of a problem are known in advance and they never change in time, it is relatively easy to build configurable hardware using programmable devices like FPGA (Field Programmable Gate Array) because the designer know how the hardware should be configured. However, for problems where designers cannot know in advance how to configure the hardware, it is required for configurable hardware to have a capability of autonomous and on-line adaptation to a given problem[17]. That is why we have implemented navigation approach , this digital implementation of *ES_GA_FL* can make use of full custom VLSI, semi custom, ASIC (Application Specific Integrated Circuits) and FPGAs (Field Programmable Gate Array). Particularly, FPGAs Implementation of *ES_GA_FL* is very attractive because of the high flexibility that can be achieved through the reprogramability nature of these circuits .

The complexity of VLSI circuits is being more and more complexes. Nowadays, the key of the art design is focused around high level synthesis which is a top down design methodology, that transform an abstract level such as the VHDL language (acronym for Very High Speed Integrated Circuits Hardware Description Language) into a physical implementation level [1.2.24.15]. In addition, the synthesis tools allow designers to realise the mainly reasons : the need to get a correctly working systems at the first time, technology independent design, design reusability, the ability to experiment with several alternatives of the design, and economic factors such as time to market . In this section , we present a new design methodology of *ES_GA_FL* based upon a VHDL description and using a synthesis tool Galileo [9]. The result is a netlist ready for place and root using the XACT[30]. The intended objective is to, realise an architecture that takes into account the parallelism, performance, flexibility and their relationship to silicon area[9].

III-1 Design methodology

The proposed design method for the *ES_GA_FL* implementation is illustrated in figure 6 as a process to follow. This status is followed by the VHDL description of the navigation approach . Then the VHDL code is passed through the synthesis tool Galileo . The result is a netlist ready for place and root using the XACT tool. At this level, verification is required before final FPGA implementation.

III-2 The *ES_GA_FL* architecture

The simplified model of *ES_GA_FL* is presented in Figure 7. Thus , we can represent it in its hardware equivalent model. The hardware model is mainly based on :

Memory circuit RAM : the data of camera sensors are stocked which are mainly the co-ordinates of milestones In effect , these points are the elements of image matrix. The depth of RAM is equal to the number of data input.
Hardware component *ES_GA_FL* : each component is composed in general into multi sub –component tasks, the most general view about it is described on the figure 8 and figure 9.
Fuzzy control : this task is to control point to point the reference trajectory obtained from hardware genetic algorithms. The GA process search for a better hardware where it is initialised . The better performance is given to fuzzy part to be controlled and treated in the next.
Optimal Path : this level gives the results of *ES_GA_FL* which is the best path suitable to navigate.

III-3 GA and FL chip

The top view of architecture of *ES_GA_FL* at level is building block for the implemented approach. The GA chip has a binary tree of training process component. The binary tree is very useful when the process is executed . A training set of the data point was generated at every generation . The processor FL in the *ES_GA_FL* chip executes the process of FL. After the optimization given, by the level GA(reference trajectory) , each component of FL calculates the degree measured data belonging to the membership function for the input variable.
The FL is based on the inference rules where the association of these rules is done among membership function of different inputs, and the processor is achieved by giving the right output after learning the navigation approach problem. We present the top view of GA and FL structure respectively in Figure 8 and Figure 9.

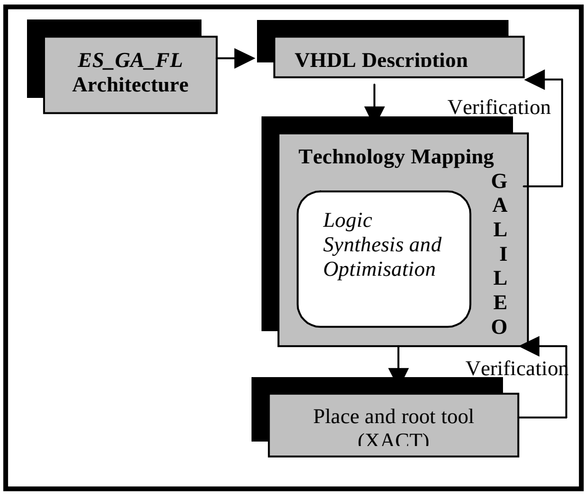

Figure 6: Design methodology of the *ES_GA_FL*

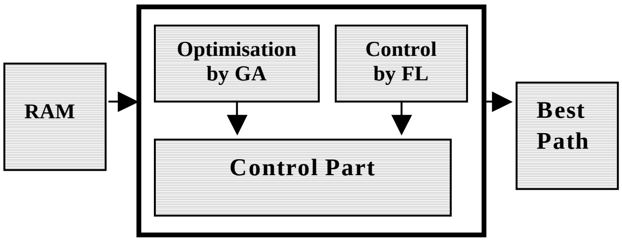

Figure 7 : Architecture of *ES_GA_FL*

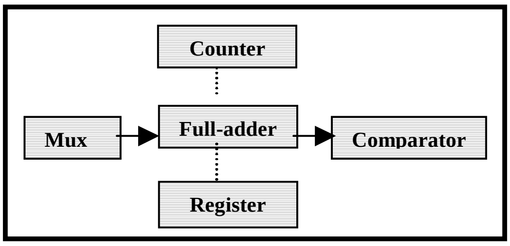

Figure 8 : The Top view of GA structure

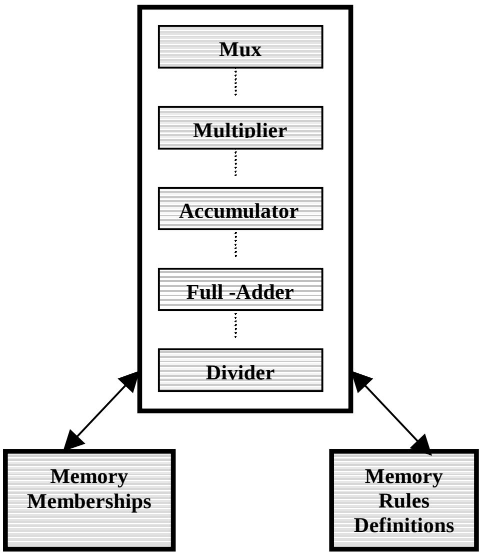

Figure 9: The top view of FL structure

IV- VHDL description of the ES_GA_FL and synthesis implementation results

IV-1 VHDL description of the ES_GA_FL

Synthesis of this design is achieved by using the VHDL language with register transfer logic (RTL) style description. The choice of VHDL comes from its emergence as an industry standard . The RTL style is used because it is well adapted for synthesis . The description of the *ES_GA_FL* begins by creating the components. As an example of component, the Figure 10 illustrates the description of the RAM where the flexibility of the design is introduced by *generic* statement.

IV-2 Synthesis and implementation results

The described methodology has been used for FPGA using the synthesis tool Galileo. At this level, and depending on the target technology, which is in our
case, the FPGA Xilinx XC4000 family, the synthesis tool proceeds to estimate area in term of CLBs (Configurable Logic Blocs). The table III presents the best synthesis results for *ES_GA_FL* .

Design	*AREA (in terme of CLBs)*
ES_GA_FL	*184*

Table III : Synthesis results

The output (XNF file) generated by the synthesis tool is passed trough the XACT place and tool .
The Figure 11 shows the resulting FPGA implementation of navigation approach as we can see the whole architecture has been into a single FPGA mapped.

```
library ieee;
use ieee.numeric_bit.all;
entity ram is
  generic( DEEP: integer := N;
  WIDTH:integer := M);
  port(wr,clk   :bit;
   adram   :in unsigned ( DEEP downto 1);
   data_in :unsigned( WIDTH downto 1);
   data_out:out unsigned(WIDTH downto 1));
end ram;
architecture behav_ram of ram is
begin
process_ram: process(clk)
 type memory is array (N downto 0) of
 unsigned(WIDTH downto 1);
 variable mem : memory;
 begin
  if clk ='1' and clk'event  then
    data_out <= ( others => '0');
    if wr ='1' then
     mem(to_integer( adram)):=data_in;
    else
    data_out <=mem(to_integer( adram));
  end if;
 end if;
end process;
end;
```

Figure 10: VHDL description of the component RAM

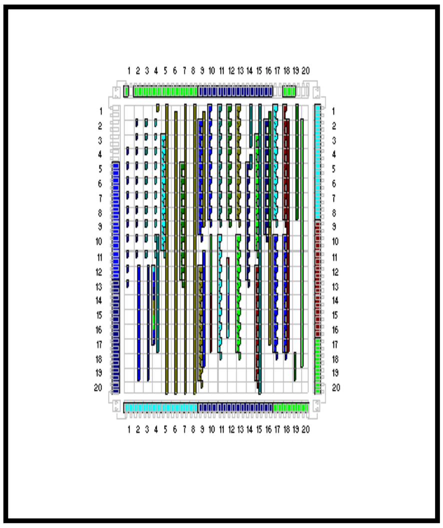

Figure 11. The top view of the *ES_GA_FL* FPGA structure

V- Conclusion

In this paper, we have presented hardware implementation of navigation approach of an autonomous mobile robot in an unknown environment. The proposed approach can deal a wide number of environment. The trained GA and FL constitute the knowledge bases of *ES_GA_FL* This system constitutes the knowledge bases allowing to recognise situation of the target localization and obstacle avoidance, respectively. To deal with the intelligent navigation control of an autonomous mobile robot , a new design methodology of *ES_GA_FL* is introduced based upon a VHDL description and using a synthesis tool Galileo. The top down of this design based on logic synthesis has allowed a single chip FPGA implementation. The proposed VHDL description has the advantage of being generic and can be changed at the user demand. The primary results of the digital navigation shows that the "intelligence- autonomy- economy" are achieved. However in the future, it is necessary to use a "micro-robot" in hostile environment and space exploration or other applications by using advanced micro-product control systems, furthermore, to reach the market with new product in the shortest possible time, and so reduced development and production time. The financial risk incurred in this development of new product can be limited so that more new ideas can be prototyped.

REFERENCES

1. Airiau, R., Bergé, J.M., Olive, V., Rouillard,J. : VHDL du langage à la modélisation, presse Polytechniques et Universitaires Romandes et CNET- ENST.

2. Airiau, R., Bergé, J.M., Olive, V. : Circuit synthesis with VHDL, Kluwer Academic Publishers, 1994.

3. Brown, S.D., Francis , R.J., Rose, J. and Vranesic, Z.G. : Field-Programmable Gate Array, Kluwer

Academic Publishers, 1997.

4. Bosacchi, B. and Masaki, I. : Fuzzy logic technology & the intelligent highway systems (IHS), in Proc. 2 nd Int. IEEE Conf.Fuzzy systems, Vol. I, San Francisco, California, 1993, pp.65-70.

5. Chuen, C.L. : Fuzzy logic in control systems : Fuzzy logic controller, Part I and Part II, IEEE Trans. On Systems and Man Cybernetics, Vol.20, N.2, March, April, 1990, pp.404-435.

6. Cummings, M. and Haruyama, S. : FPGA in the soft radio, IEEE Communication Magazine, 0163-1999, IEEE, Communications Magazine,pp.108-112.

7. Elithorn, A., Banerji, R. : Artificial & human Intelligence, Elsevier Science Publihers, B.V. NATO, 1984.

8. Filho, J.L.R., Treleaven, P.C. and Alippi, C. : Genetic – Algorithm Programming Environments, 1994, IEEE, pp.28-73.

9. GALILEO HDL Synthesis Manual, Exemplar Logic,1995.

10. Gleizes, M.P. and Glize, P.: Les systèmes multi-experts, Edition – Hermès, Paris,1990.

11. Gonzàlez, A. and Pérez, R. : SLAVE : A genetic learning System Based on an Iterative , Approach, IEEE, Trans on Fuzzy systems, Vol 7, N.2, April 1999, pp.176-191.

12. Hewett, M. and Hewett , R. : A language and Architecture for Efficient Blackboard Systems, in Proc, 9.th.Conf.Artificial Intelligence for applications, Orlando, Florida, March 1-5, 1993, pp.34-40.

13. Keirsey, D.M., Koch, E., Mckisson, J., Meystel, A.M. and Mitchell, J.S.B. : Algorithm for Navigation for a Mobile Robot, 1984,IEEE,pp. 574-583.

14. Kujawski, C. : Deciding the behaviour of an autonomous mobile road vehicle, in Proc.2nd, Int.IFAC Conf, Intelligent Autonomous Vehicles, Finlind,1995,pp.404-409.

15. Legenhausen, J., Wade, R., Wilner, C. and Wilson, B. : VHDL for programmable logic, Addison- Wesley, 1996.

16. Niegel, W. : Methodical structuring of knowledge used in an intelligent driving system, in Proc.2nd Int.IFAC Conf. Intelligent Autonomous Vehicles, Finland, 1995, pp.398-403.

17. Murakawa, M., Yoshizawa, S., Kajitani, I., Yao, X., Kajihara, N., Iwata, M. and Higuchi, T. : The GRD chip : genetic reconfiguration of DSPs for Neural Network Processing, IEEE Trans on Computers, Vol.48, N.6, June 1999, pp.628-639.

18. Moreno, L ., Puente, E.A and Salichs, M.A. : World modelling and sensor data fusion in a non static environment : application to mobile robots, in Proc.Int.IFAC Conf.Intelligent Components and Instruments for control Applications, Malaga, Spain, 1992, pp.433-436.

19. Masmoudi, N., Hachicha, M. and Kamouni, L. : Hardware design of programmable fuzzy controller on FPPGA, in Proc.Int IEEE Conf, Fyzzy Systems, August 22-25, 1999, Seoul, Korea, ppS.III-1675-III-1679.

20. Medsker, L.R.: hybrid intelligent systems, Kluwer Academic Publishers, 1995.

21. Mitra, S.and Pal, S.K. : Fuzzy multi layer percetron, Inferencing and rule generation. IEEE Trans on Neural Networks, Vol.6, N.1, 1995, pp.51-63.

22. Pedrycz, W.: Relevancy of fuzzy models, Information Sciences 52(1990), 285-302.

23. Pedrycz, W. Fuzzy control and Fuzzy systems & sons, Jhon Wiley, 1988.

24. Pellerin, D. and Taylor, D. : VHDL made easy, Prentice Hall PTR, 1997.

25. Shih-Sen, C. : Fixed point theorems for fuzzy mapping, Fuzzy sets and systems17 (1985), 181-187.

26. Schilling, K. and Jungius, C. : Mobile robots for planetary explorations, in Proc.2nd Int.IFAC Conf.Intelligent Autonomous Vehicles, Finland, 1995, pp.110-120.

27. Srinivas, M. and Patnaik, L.M. : Genetic Algorithms : a survey, 1994 IEEE, Computer, pp.17-26.

28. Spiegel, M.R.: Séries Schaum : probabilité et statistique, Publication Scientifique et littéraires 05002 GAP, 1983.

29. Yager, R.R. : A characterization of the extension principle, Fuzzy Sets and Systems 18 (1986), 205-217.

30. XACT user manual, © 1995.

31. Zadeh, L.A. : The calculus of Fuzzy if/ then rules, AI expert, March 1992,pp.23-27.

32. Zadeh, L.A. : Fuzzy logic = computing with words, IEEE Trans on Fuzzy Systems, Vol.4, N.2, May 1996, pp.103-111.

33. Zadeh, L.A. : Fuzzy sets, Inform & Control, 1965,pp.338-353.