Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 22 (2016) 600 – 605

Procedia Technology

9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu-Mures, Romania

Adaptive Neuro-Fuzzy Structure Based Control Architecture

Tibor Tamas[a], Szabolcs Hajdu[a], Sándor Tihamér Brassai[a,*]

[a]Sapientia Hungarian University of Transylvania, 1C, Corunca 547367, Romania

Abstract

The purpose of this paper is to present a practical application of a Sugeno model based adaptive neuro-fuzzy architecture. The main challenge of the project was to realize the real-time control of the system and the real-time parameter adaptation of the controller. The neuro-fuzzy controller module is implemented using High Level Synthesis technique, and it's integrated into a microprocessor based architecture on a System on Chip (SoC) type integrated circuit. The architecture is used to control a two degrees of freedom system, which is composed of two horizontal arms attached to a linear bearing. The bearing is running on a vertical beam so the system can execute a vertical motion and a rotary motion around the vertical axle. The positions according to the two degrees of freedom are determined using an ultrasonic and a magnetic sensor.

Keywords: neuro-fuzzy; FPGA; HLS; adaptive embedded control.

1. Introduction

Since the development of the first theories combining the fuzzy systems and the artificial neural networks [1], the main obstacle for the practical applications was the lack of proper hardware to realize real time control systems. Nowadays there are more possibilities for the implementation of a real time working neural network based system. One of the possibility is to use microprocessors with high operating frequency, but in case of dedicated embedded architectures in most of cases there are power supply constraints, and must be also solved the problem of cooling the circuit. The other possibility is to realize dedicated hardware based on the structure of the neural network, in which

* Corresponding author. Tel.: +40 265 208 170; fax: +40 265 206 211.
E-mail address: tiha@ms.sapientia.ro

doi:10.1016/j.protcy.2016.01.126

the operations could be parallelized. FPGA circuit based neuro-fuzzy implementations are presented in [2] [3] [4] works. With parallelized operations can be achieved fast enough output calculation to operate the circuit in real time. In our project we used a Zynq type SoC integrated circuit which alongside the programmable logic also contains a powerful dual-core ARM processor. In our project, because it is a full physical system, with sensors, control unit, communication modules and actuators, we are distributing the tasks between different modules. There are communication interfaces for the sensors, and for the RF communication module, there is a PWM signal generator module, BRAM memory controllers and memory blocks, and the ANFIS controller block, all these implemented into the programmable logic of the Zynq SoC integrated circuit. On the other hand, on the ARM hard-processor is running the software, that controls all these modules, and also in the processors program it's implemented the training algorithm of the ANFIS controller. The modules implemented in the programmable logic can operate in parallel, while the software running on the ARM processor is executed sequentially, but in a much higher clock frequency. This strategy was also used in the [5] work, where a PWM-ANFIS (Piecewise Multilinear – Adaptive Neuro-Fuzzy Inference System) is used, with the controller part implemented in the programmable logic of an Altera SoC and the training algorithm is running on the hard processor.

2. Hardware architecture of the system

The purpose of building the mechanism used in this work was to make a suitable test bench for an ongoing project which is an autonomous quadcopter type UAV. Some other neuro-fuzzy based quadcopter control systems are presented in [6] [7] works. This structure it's a two degrees of freedom system and is composed of a plastic profile, with a linear bearing running on a 2 meters long hardened steel shaft. The plastic profile has attached two arms with a brushless motor with propellers on each of them (Fig. 1).

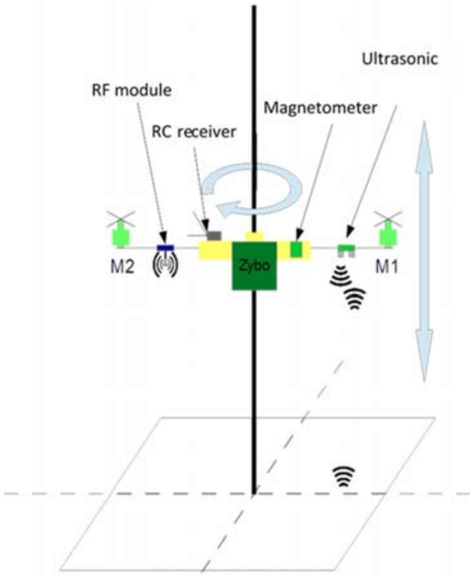

Fig. 1. Physical structure

The ultrasonic sensor is used to determine the altitude of the device and the magnetometer for the determination of the horizontal orientation. The moves simulated with the device are the cases which occurs when a quadcopter is landing or taking-off. A communication module with radio frequency is attached to the system which sends measurement data to another FPGA board. The data received is plotted by the System Generator program running on the second FPGA board. There is also an RC receiver on the device, through what we can give reference signal to the system. All these components are connected to the Zybo SoC board.

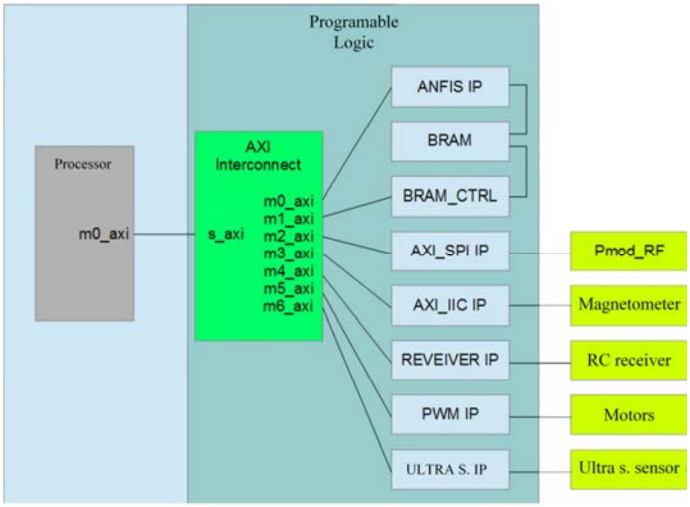

Fig. 2. Hardware architecture

On the Fig. 2 we can see the implemented hardware architecture, containing the sensor interfaces, the communication modules and the controller IP core. The ANFIS IP core is a module developed in C language and synthetized to VHDL (VHSIC Hardware Description Language) language using Xilinx Vivado HLS tool. This is a relatively new tool that allows to develop a hardware module on behavioral level. The advantages are the much shorter design time, than in case of other techniques, the availability of many optimizing directives and the easy interface implementation offered by the tool. The ANFIS controller has a Sugeno model based structure [1] implemented as a five layer artificial neural network which has two inputs and one output. For the presented application are used five bell-shaped membership functions on each of the two inputs. The IP has an AXI interface through which it's controlled by the processor, and has a BRAM interface to connect to the memory containing the parameters. As we can see on the Fig. 2. the ANFIS IP has attached a port of a dual-port BRAM block and the other port is connected to the AXI interconnect through a BRAM controller. The BRAM memory contains the parameters of the ANFIS controller, which are the parameters of the membership functions and the p, q and r parameters of the third layer.

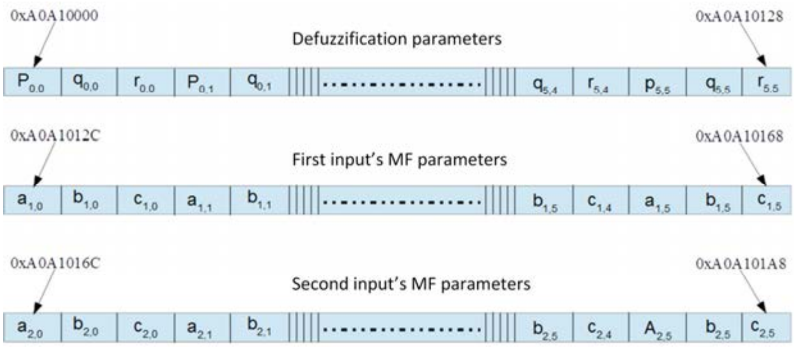

Fig. 3. The placing of the parameters in the BRAM memory

The a, b and c are the parameters of the bell-shaped membership functions (1), as we can see on the Fig. 3. the parameters for the two inputs are stored in the memory consecutively. The p, q and r parameters are used in the deffuzification phase and these are the tuneable parameters of this system.

$$\mu = \frac{1}{(1+\left|\frac{x-c}{a}\right|^{2b})} \quad (1)$$

$$Y = \sum_{j=1}^{5}\sum_{i=1}^{5}(w_{i,j} * z_{i,j}(x_1, x_2)) = w_{i,j} * (p_{i,j}x_1 + q_{i,j}x_2 + r_{i,j}),\ i=1\ldots5, j=1\ldots5 \quad (2)$$

where x_1 and x_2 are the two inputs of the controller, $w_{i,j}$ is the weight calculated in the previous layer

The controller module is designed in that way, to be possible the implementation of multiple controllers with a single IP core by placing multiple parameter sets into the BRAM block. With an input register value can be specified which of the controllers to be used. For the controller structure presented there are stored in the memory two parameter sets representing the parameters of the two controllers. The membership functions are covering the [0, 1024] interval, so the input values must be scaled according to this. In the Table 1 can be seen the parameter values set to the membership functions. The p, q and r tunable parameters are initialized to 0.

Table 1 Parameters of the membership functions

a	b	c
128	3	0
128	3	256
128	3	512
128	3	768
128	3	1024

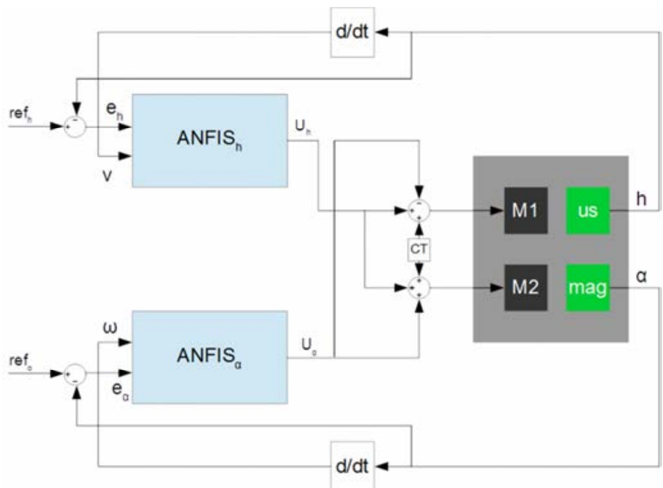

Fig. 4. The control structure of the physical system

The Fig. 4 illustrates the logical control structure of the 2 DOF system. The inputs of the first controller are the altitude error (eh) and the speed (v). The inputs of the second controller is the angle error (eα) and the angular velocity (ω). The control signals of the motors are calculated with the following formulas:

$$M1 = U_{ct} + U_h + U_\alpha \quad (3)$$

$$M2 = U_{ct} + U_h - U_\alpha \quad (4)$$

where M1 and M2 are the duty cycles of the two PWM signals controlling the motors. Uct is the maximum value that doesn't lift the device. Uh is the output of the first controller, which controls the altitude and Uα is the output of the second controller which controls the angle. The angle is calculated from the measurements of the magnetometer with the following formula:

$$\alpha = arctg2(magY, magZ) * \frac{180}{\pi} \tag{5}$$

3. Training of the parameters

The training algorithm is executed by the processor and executes a real-time parameter adaptation. The parameter sets are placed in the BRAM starting from the base address of the memory. For the parameter tuning the ANFIS IP core is returning to the processor the indices of the active membership functions on the two input spaces and the weight corresponding to each pair of active membership functions. The indices determine the location in the memory of the parameters that must be adapted. The p, q and r parameters are adapted with the following gradient descend based algorithm:

$$p = p + \eta * e * x_1 * \overline{w} \tag{6}$$

$$q = q + \eta * e * x_2 * \overline{w} \tag{7}$$

$$r = r + \eta * e * \overline{w} \tag{8}$$

where η is the training coefficient, e is the error calculated from the output and the reference value, x_1 and x_2 are the inputs of the controller and $\overline{w}$ is the normalized weight corresponding to the used membership function indices. The training process is highly influenced by the value of the training coefficient.

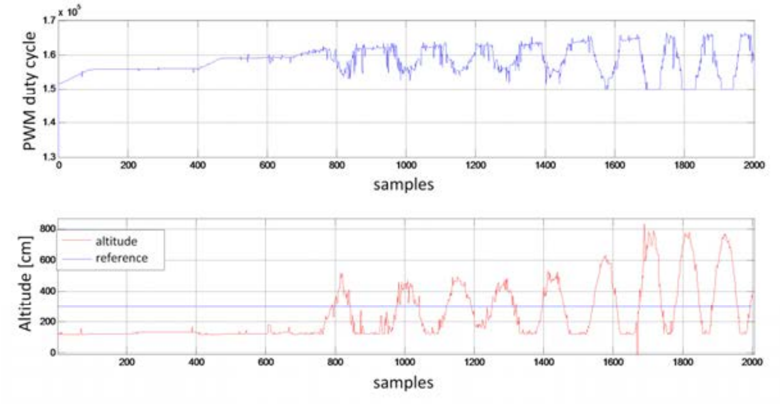

Fig. 5. Training process with too high training coefficient

A critical parameter of the training is the training coefficient. On the Fig. 5 we can see a measurement during the training process, where the value of η is too high and the parameters of the third layer are changing too suddenly and the system begin to oscillate. On the first diagram is plotted the control signal of the motors and on the second diagram the measured altitude and the reference value. On the Fig. 6 we can see a measurement with smaller training coefficient, and we can see that the control signal is growing slower and the altitude stabilizes near the reference value.

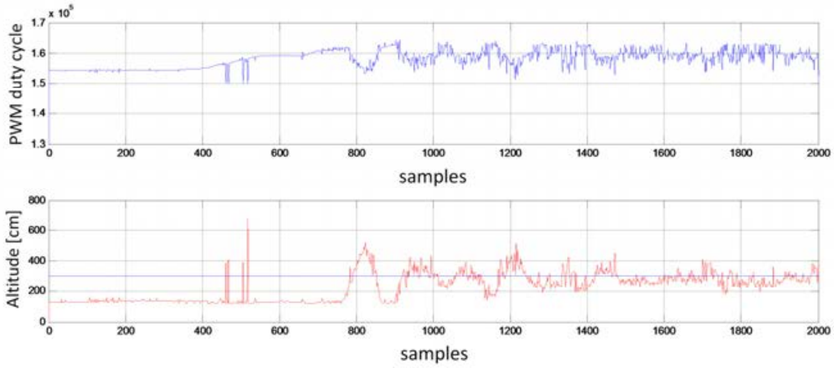

Fig. 6. Training process with proper training coefficient value

4. Conclusions

We have presented a practical application of an adaptive neuro-fuzzy controller. The controller is implemented using the high level synthesis technique, and it's integrated into a microprocessor based architecture. The controlled device is two degrees of freedom system, which is equipped with sensors for altitude and orientation determination. The present project is a base part of a more complex autonomous quadcopter developing project. Reasoning from the measurement we can conclude that the developed ANFIS controller could be a reliable solution for real time control of non-linear systems.

Acknowledgements

The research presented in this paper was supported by the Sapientia Foundation – Institute for Scientific Research.

References

[1] Jang JSR. "Adaptive-Network-Based Fuzzy Inference Systems," IEEE Transaction on Systems, Man and Cybernatics, pp. 665-685, 1993.

[2] Echanobe J, del Campo I., Basterretxea K., Martinez MV, Doctor F. "An FPGA-based multiprocessor-architecture for intelligent environments," Microprocessors and Microsystems, vol. 38, pp. 730-740, 2014.

[3] Elloumi M, Krid M, Masmoudi DS. "Implementation of Neuro-Fuzzy System based image edge detection," Proceedings of 2013 IFIP/IEEE 21st International Conference on Very Large Scale Integration (VLSI-SoC), 2013.

[4] Cermak P, Pokorny M. "The Fuzzy-Neuro development system FUZNET," Methods and Models in Automation and Robotics (MMAR), 2013 18th International Conference on, pp. 26-29, 2013.

[5] del Campo I, Echnaobe J, Bosque G, Tarela JM. "Hardware/Software Implementation of an Adaptive Neuro-Fuzzy System," IEEE TRANSACTIONS ON FUZZY SYSTEMS, vol. 16, no. 3, pp. 761-779, 2008.

[6] Rezazadeh S, Ardestani MA, Sadeghi PS. "Optimal attitude control of a quadrotor UAV using Adaptive Neuro-Fuzzy Inference System (ANFIS)," Control, Instrumentation, and Automation (ICCIA) 2013 3rd International Conference on, pp. 219-223, 2013.

[7] Bhatkhande P, Havens TC. "Real time fuzzy controller for quadrotor stability control," Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on, pp. 913-919, 2014.

[8] Lu HC, Chang MH, Tsai CH. "Adaptive self-constructing fuzzy neural network controller for hardware implementation of an inverted pendulum system," Applied Soft Computing 2011; 11: 3962-3975.