UDC 621.3 https://doi.org/10.20998/2074-272X.2023.3.04

A. Aib, D.E. Khodja, S. Chakroune

Field programmable gate array hardware in the loop validation of fuzzy direct torque control for induction machine drive

Introduction. *Currently, the direct torque control is very popular in industry and is of great interest to scientists in the variable speed drive of asynchronous machines. This technique provides decoupling between torque control and flux without the need to use pulse width modulation or coordinate transformation. Nevertheless, this command presents two major importunities: the switching frequency is highly variable on the one hand, and on the other hand, the amplitude of the torque and stator flux ripples remain poorly controlled throughout the considered operating speed range. The* ***novelty*** *of this article proposes improvements in performance of direct torque control of asynchronous machines by development of a fuzzy direct torque control algorithm. This latter makes it possible to provide solutions to the major problems of this control technique, namely: torque ripples, flux ripples, and failure to control switching frequency.* ***Purpose****. The emergence of this method has given rise to various works whose objective is to show its performance, or to provide solutions to its limitations. Indeed, this work consists in validation of a fuzzy direct torque control architecture implemented on the ML402 development kit (based on the Xilinx Virtex-4 type field programmable gate array circuit), through hardware description language (VHDL) and Xilinx generator system. The obtained* ***results*** *showed the robustness of the control and sensorless in front of load and parameters variation of induction motor control. The research directions of the model were determined for the subsequent implementation of results with simulation samples.* References 19, tables 5, figures 26.

Key words: **fuzzy control, field programmable gate array, Xilinx system generator, direct torque control, power system.**

Вступ*. В даний час пряме управління моментом дуже популярне в промисловості і викликає великий інтерес у вчених у галузі частотно-регульованого приводу асинхронних машин. Цей метод забезпечує розв'язку між керуванням моментом, що крутить, і магнітним потоком без необхідності використання широтно-імпульсної модуляції або перетворення координат. Тим не менш, ця команда представляє дві основні незручності: з одного боку, частота комутації сильно варіюється, а з іншого боку, амплітуда пульсацій моменту і потоку статора залишається погано контрольованою у всьому діапазоні робочих швидкостей.* ***Новизна*** *цієї статті пропонує поліпшення характеристик прямого керування моментом, що крутить, асинхронних машин шляхом розробки нечіткого алгоритму прямого управління моментом, що крутить. Останнє дозволяє вирішити основні проблеми цього методу управління, а саме: пульсації моменту, що крутить, пульсації потоку і нездатність контролювати частоту перемикання.* ***Мета****. Поява цього методу породило різні роботи, метою яких є показати його ефективність чи запропонувати рішення стосовно його обмежень. Дійсно, ця робота полягає у перевірці нечіткої архітектури прямого управління моментом, що крутить, реалізованої в наборі для розробки ML402 (на основі схеми Xilinx Virtex-4 з програмованою користувачем вентильною матрицею), за допомогою мови опису обладнання (VHDL) та генераторної системи Xilinx.* ***Отримані результати*** *показали робастність керування та безсенсорного керування при зміні навантаження та параметрів керування асинхронним двигуном. Визначено напрями дослідження моделі для подальшої реалізації результатів на імітаційних вибірках.* Бібл. 19, табл. 5, рис. 26.

Ключові слова: **нечітке управління, програмована користувачем вентильна матриця, генераторна система Xilinx, пряме управління моментом, що крутить, система живлення**.

Introduction. Direct Torque Control (DTC) was realized by Takahashi and Depenbrock [1, 2] is more and more popular and it interests many scientists and industrialists in the field of variable speed applications [3, 4].

However, this strategy has two major drawbacks: on the one hand, the switching frequency is highly variable and on the other hand, the amplitude of the ripples of torque and of stator flux is poorly controlled over the entire speed range of the operation envisaged [5]. It should be noted that torque ripples generate additional noise and vibration and therefore cause fatigue in the rotating shaft [6].

To further reduce the impact of these phenomena on the service life of electric actuators, it is believed that intelligent techniques can provide an improvement. In terms of real-time management of managed applications based on intelligent techniques, there are new hardware design solutions such as Field Programmable Gate Array (FPGA) or application specific integrated circuit [7, 8]. These reconfigurable circuits are available and can be used as digital targets for implementation of control algorithms in a single component [9].

The advantages of such an implementation are multiple: reduction of execution time by adopting parallel processing, rapid prototyping of the numerical control on FPGA. The confidentiality of architecture and possibility of application of intelligent controls make use of techniques which are more cumbersome in terms of computation time and the improvement of the quality of the control of electric machines [10, 11].

Evolution of micro computing, semiconductor technology and availability of rapid control means such as digital signal processor, reconfigurable circuits (FPGA). Today allows the scientific community to carry out very complex controls while taking into account the non-linearity of mathematical model of asynchronous machine [12, 13].

The goal of the paper is to evaluate the performance of the use of a fuzzy DTC inference system versus the classic DTC based on hysterized comparators, for the control of induction machines (IMs) based on a FPGA using available academic tools (MATLAB/Simulink, Xilinx system generator, Xilinx ISE, and ModelSim).

Basic calculation relationships and assumptions. The subject of DTC is controlling the torque and stator flux of asynchronous machine by applying several voltage vectors through a voltage inverter.

The control is generally carried out by a hysteresis controller. The purpose of control is to keep controlled variables within a specified hysteresis band [6]. The controller provides the necessary switching pulses to the inverter to generate the optimum voltage vector to supplies the IM for a defined operating condition. The IM model is used with measured variables to estimate stator flux and electromagnetic torque required by control diagram (Fig. 1), where S_a, S_b, S_c are the Boolean switching commands; ω is the speed for reference speed ω_{ref}; E_φ, E_{T_e} are the flux and torque error respectively; T_e is the electromagnetic torque; T_{eref} is the reference electromagnetic torque; I_{sa}, I_{sb} are the stator currents in the *abc* reference frame; ϕ_{sref} is the reference stator flux; $|\varphi_s|$ is the stator flux magnitude; θ_{ϕ_s} is the stator flux angle.

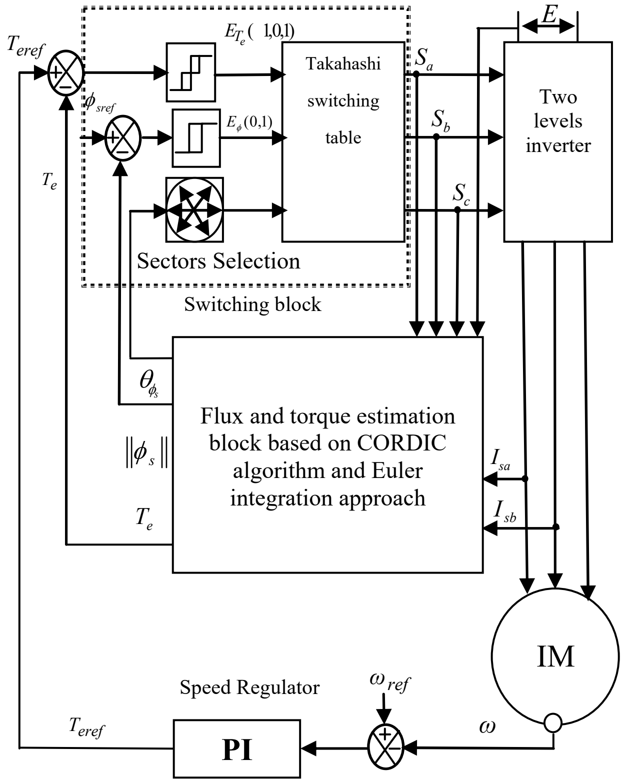


Fig. 1. DTC based IM control structure

The most significant element that can guarantee satisfactory DTC performance is stator flux estimator and torque estimator [12]. In this work we used optimized estimator developed in [14].

Fuzzy DTC based IM control structure. The research theme developed in this work mainly concerns the exploitation of new technological solutions to implement an intelligent control based on the DTC of an asynchronous machine around a hardware environment based on a FPGA.

This implementation is mainly aimed at reducing ripples at the level of electromagnetic torque and stator flux. In this part, two hysteresis regulators and Takahashi switching table (Table 1) will be replaced by a fuzzy controller. Figure 2 shows the control structure of fuzzy DTC based IM.

Table 1

Takahashi switching table

		$V_i = (S_a, S_b, S_c)$					
ϕ, T_e, N		N=1	N=2	N=3	N=4	N=5	N=6
ϕ=1	T_e=1	(1,1,0)	(0,1,0)	(0,1,1)	(0,0,1)	(1,0,1)	(1,0,0)
	T_e=0	(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)
	T_e=–1	(1,0,1)	(1,0,0)	(1,1,0)	(0,1,0)	(0,1,1)	(0,0,1)
ϕ=0	T_e=1	(0,1,0)	(0,1,1)	(0,0,1)	(1,0,1)	(1,0,0)	(1,1,0)
	T_e=0	(0,0,0)	(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)	(1,1,1)
	T_e=–1	(0,0,1)	(1,0,1)	(1,0,0)	(1,1,0)	(0,1,0)	(0,1,1)

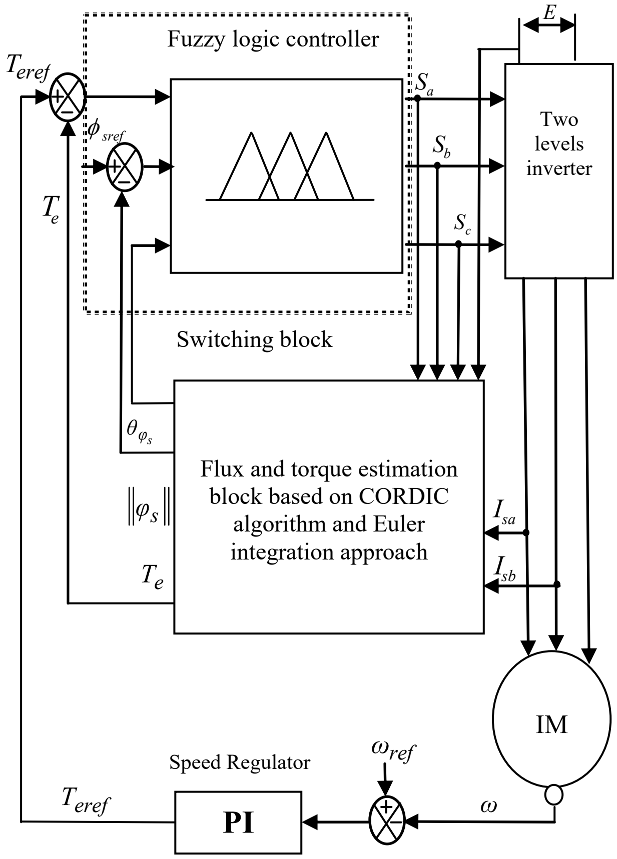


Fig. 2. Fuzzy DTC based IM control structure

Fuzzy DTC control implementation. The hardware implementation of a fuzzy inference system consists in implementing 3 phases of a fuzzy logic regulation: fuzzification, fuzzy inferences and defuzzification. This principle is represented by Fig. 3.

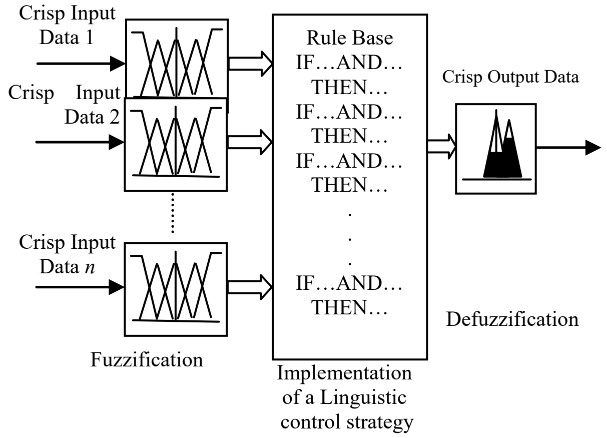


Fig. 3. Components of a fuzzy inference system

Description of fuzzification module. Fuzzification is the process of converting input data into fuzzy linguistic values. In literature, there are two material solutions to determine the degree of membership of a fuzzy set from a membership function.

The first solution is the memory-oriented approach, as the name suggests, for each finite number of inputs, the output values are calculated offline then they will be saved in memory. The advantage of this solution is that it is easy to change a membership function. The second solution is the calculation-oriented approach, only the characteristics of the membership functions are saved in a memory in order to simplify the on-line calculation of output values of each membership function. For the case of triangular membership functions, their characteristics are: the center of the triangle «*c*» and the slope «*a*».

The hardware implementation of this solution is a combinatorial circuit which can include adders, subtractors, multiplexers, multipliers and most of the time a control unit.

In this study, we adopt the memory-oriented approach. Indeed, each linguistic input/output variable is represented by tables, a table for the degree of membership of each linguistic value. These tables are implemented in hardware by memory blocks ROMs addressable by a single input, such as the memory boxes which contain the degree of membership of linguistic value. However, the memory address space gives an image on the universe of discretized speech for example for a universe of normalized speech [0, 1] discretized in 64 points, we therefore use an address space [0: 63].

The membership functions of flux error, torque error, sectors and output vectors are illustrated by the following Fig. 4-7.

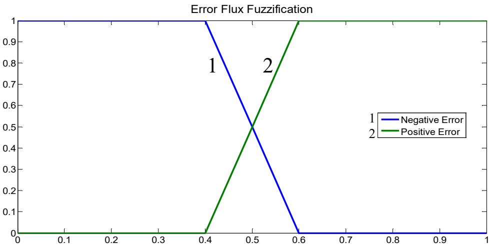


Fig. 4. The fuzzification membership functions of the flux error

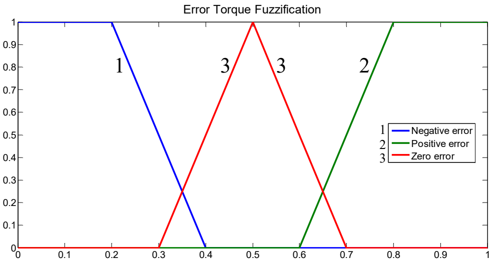


Fig. 5. Fuzzification membership functions of the torque error

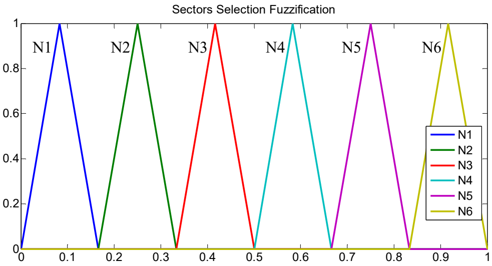


Fig. 6. Fuzzification membership functions of the angle

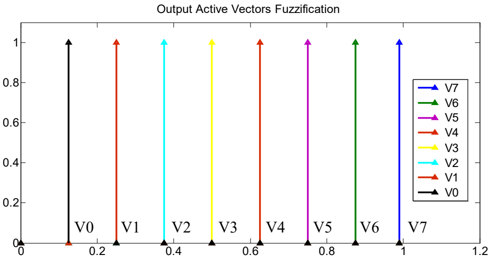


Fig. 7. Membership functions of the output

The hardware implementation of these functions is presented by hardware architectures in Fig. 8-11.

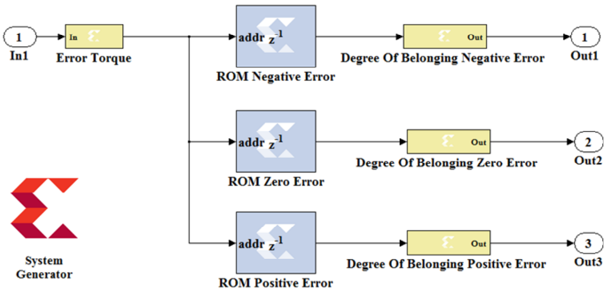


Fig. 8. Hardware architecture of the «Torque Error» linguistic variable

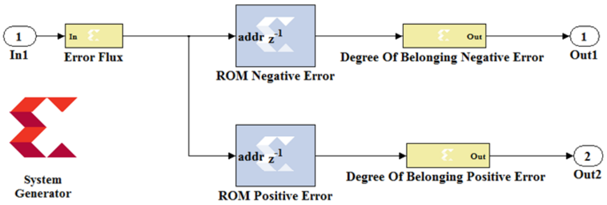


Fig. 9. Hardware architecture of the «Flux Error» linguistic variable

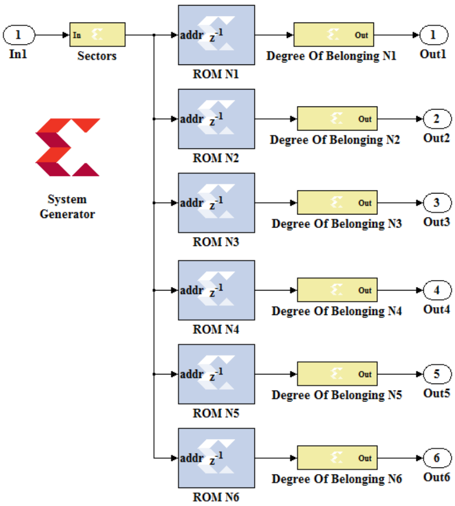


Fig. 10. Hardware architecture of the «Sectors» linguistic variable

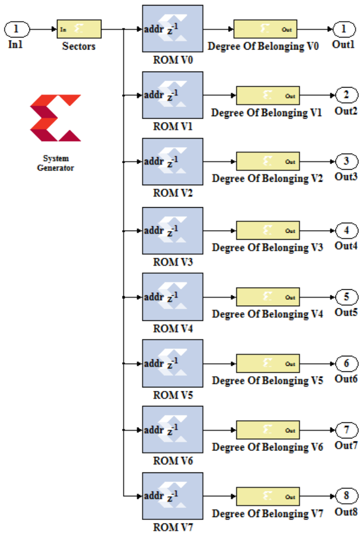
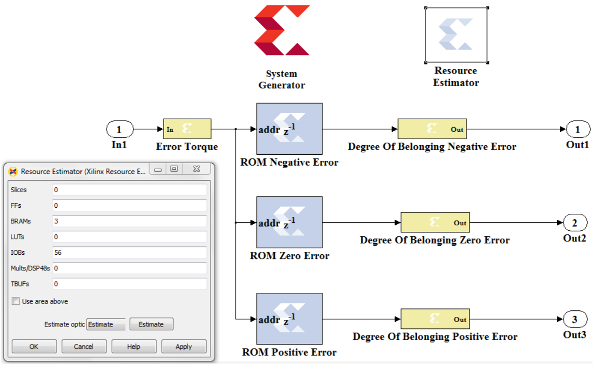


Fig. 11. Hardware architecture of the «Output vector» linguistic variable

The Xilinx resource estimator tool is used to estimate the hardware resources needed to implement each linguistic variable. Figure 12 shows the estimated resources for the linguistic variable «Torque Error».



Slices 0
FFs 0
BRAMs 3
LUTs 0
IOBs 56
Mults/DSP48s 0
TBUFs 0
□ Use area above
Estimate optic Estimate

Fig. 12. Hardware resources consumed by the linguistic variable «Torque Error»

Rule inference and rule evaluation. Hardware description of fuzzy inference module is shown in Fig. 13. This module accepts as input three blocks of fuzzification module, the rule selector block allows building the rule base formed of 36 rules. This basis is obtained by making all the possible combinations between two fuzzy values of flux error, three fuzzy values of torque error and six fuzzy values of stator flux angle.

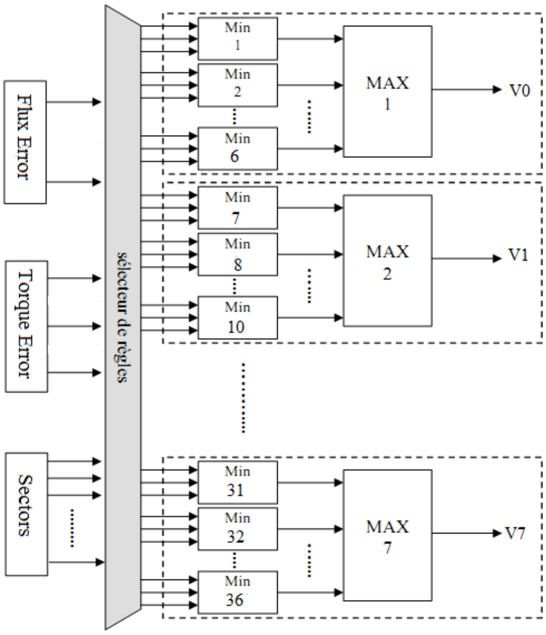


Fig. 13. Architecture of «fuzzy inference» module

The hardware implementation on «Xilinx System Generator» of operators (min/max) with 2 inputs is done by a comparator and a 2-1 multiplexer. Figure 14 shows the wiring of min/max functions.

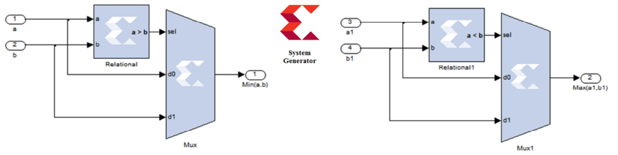
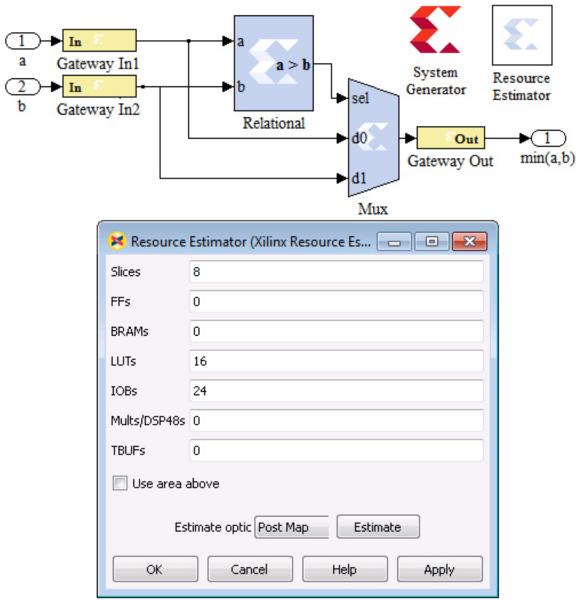


Fig. 14. Implementation of min/max functions on XSG

Using the resource estimator tool allows us to estimate the hardware resources consumed by a two-entry min operator (Fig. 15).

For operators (min/max) that have more than 2 inputs, 2 inputs (min/max) operators are used to implement these operators. For example, to implement an operator (min) with 3 inputs, we use 2 operators (min) with 2 inputs (Fig. 16).



Slices 8
FFs 0
BRAMs 0
LUTs 16
IOBs 24
Mults/DSP48s 0
TBUFs 0
□ Use area above
Estimate optic Post Map

Fig. 15. Hardware resources consumed by a two-entry min operator

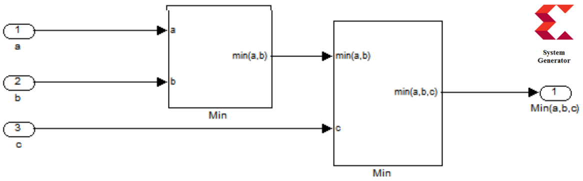


Fig. 16. Implementation of a 3-entry min function

Composition of the rules. If several rules can be activated simultaneously and recommend actions with different degrees of validity on same output, we consider that the rules which are linked by an operator OR (Fig. 17) $\mu_{By} = \max[\mu_{Bi}(y)]$ $i \in \{\text{indices of activated rules}\}$.

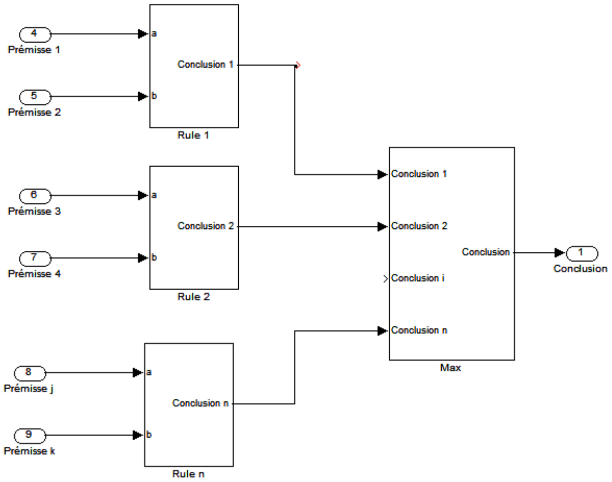


Fig. 17. Implementation of a composition of rules for an output

Description of defuzzification module. The hardware description of defuzzification module is carried out by a MAX operator as shown in Fig. 18. The inputs of this module are the outputs of the inference motor module. As it was specified in the preceding paragraph, the V_i (i = 0.7) correspond to the degree of activation of voltages V. The output of this block, representing the output of the whole fuzzy block, corresponds to the voltage that it must be applied to the terminals of the machine through the inverter. The VHDL architecture of MAX defuzzification module is implemented by an assignment in the competitive mode:

```
S135 <= 001 when V1=max
else 010 when V2=max
else 011 when V3=max
else 100 whenV4=max
else 101 when V5=max
else 110 when V6=max
else 111 when V7=max
else 000
```

Figure 18 shows a description of VHDL block used to implement the maximum defuzzification method.

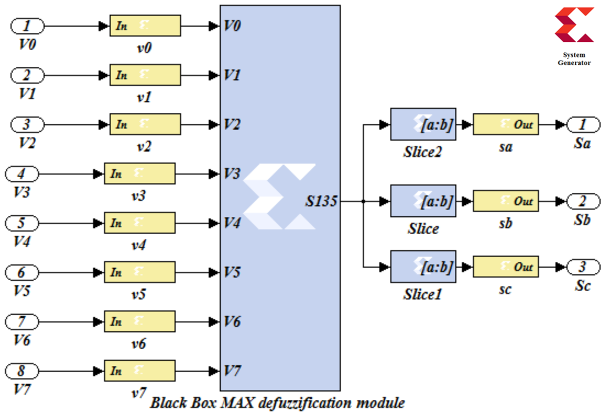


Fig. 18. VHDL MAX defuzzification module

Simulation of fuzzy DTC control of asynchronous machine. This phase involves the integration of a fuzzy inference system into DTC control algorithm. In this part, similarly to simulation with MATLAB, we will simulate using Xilinx generator system the architecture of the conventional DTC control and fuzzy DTC control. The fuzzy DTC control is applied to an IM whose specifications are given in Table 2.

Table 2

IM parameters

Nominal voltage U_n, V	220
Nominal current I_n, A	2.35
Mechanical power P_n, kW	1.08
Nominal speed N, min^{-1}	1430
Supply frequency f, Hz	50
Stator resistance R_s, Ω	10
Rotor resistance R_r, Ω	6.3
Stator self inductance L_s, H	0.4642
Rotor self inductance L_r, H	0.4612
Mutual inductance L_m, H	0.4212
Moment of inertia J, kg·m^2	0.02
Pole pairs number p	2

The structure of conventional DTC control is illustrated in Fig. 1. Regarding fuzzy DTC control shown in Fig. 2, we will replace the blocks of hysteresis comparator modules and DTC control selection table by a fuzzy inference system that we have built and tested. To compare the both control approaches, we simulated these modules on MATLAB / Simulink with Xilinx generator system. The results are illustrated in Fig. 19.

Note that there is an improvement in the electromagnetic flux and torque obtained by the fuzzy DTC control compared to that obtained with the conventional DTC control with a significant reduction in ripples. Table 3 shows the performance in terms of resource consumption, obtained during the implementation of architecture of fuzzy DTC control on VIRTEX 4 FPGA given by architecture presented in Fig. 2.

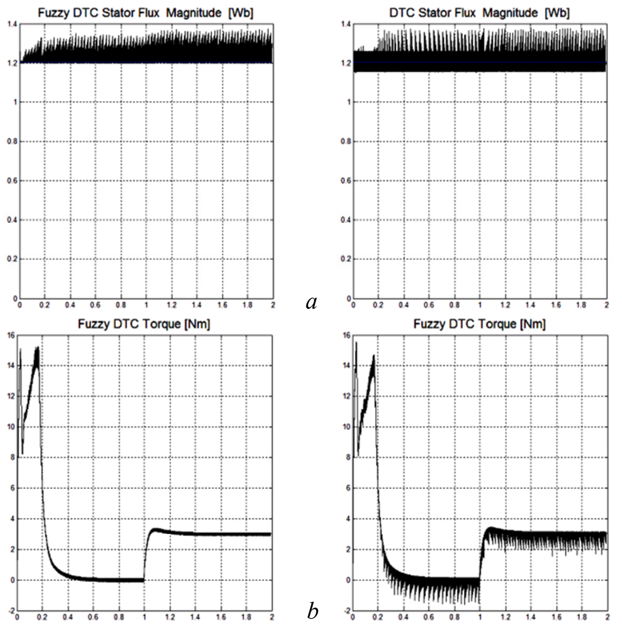


Fig. 19. Electromagnetic torque and stator flux obtained by XSG simulator for DTC (*a*) and fuzzy DTC (*b*)

Table 3

Resources used on FPGA circuit by fuzzy DTC control algorithm

Target device: ML402 Virtex-4 xc4vsx35-10ff668			
Logic utilization	Used	Available	Utilization
Number of Slice Flip Flops	1,365	30,72	4 %
Number of occupied Slices	1,620	15,36	10 %
Total Number of 4 input LUTs	2,453	30,72	7 %
Number of bonded IOBs	58	448	12 %
Number of FIFO16/RAMB16s	12	192	6 %
Number of BUFG/BUFGCTRLs	4	32	12 %

We note that the proposed architecture optimizes the use of the hardware resources of FPGA card (10 % of Slices and 7 % of LUTs), moreover this architecture considerably reduces the logical components used compared to architectures presented in [15, 16].

The maximum clock frequency is set by the synthesis tool equal to 231.64 MHz, which corresponds to a minimum period of 4.317 ns. In contrast, in [16] the maximum clock frequency is 54 MHz using DSPACE (Digital Signal Processing and Control Engineering). In [17] the minimum period is equal to 50 ns. We see that execution time is too long compared to FPGA due to sequential processing of DSPACE.

Table 4 presents the hardware resources consumed in this architecture compared to previous work in the same research axis.

Table 4

Comparison of the resources consumption

Logic utilization	References				
	[7]	[18]	[19]	[14]	Proposed
FPGA device family	Xilinx Virtex-4	Altera DE-115	Altera CYCLONE II	Xilinx Virtex-4	Xilinx Virtex-4
Embedded multiplier 9-bit elements	–	80	57	–	–
Total logic elements	10.346	6.931	3.256	2.909	2.836
Total combinational functions	18.594	6.491	2.549	7.411	7.686

Validation of hardware architecture of proposed fuzzy controller. After simulation step, the proposed hardware architecture of fuzzy controller was validated by co-simulation hardware on the ML402 target peripheral equipped with a VIRTEX4 FPGA circuit.

This step is dedicated to implement the control algorithms on a development board integrating an FPGA component. It is mainly intended for the verification and validation of digital implementation of control algorithms on FPGA targets in «Hardware in the loop» simulation environment as shown in Fig. 20.

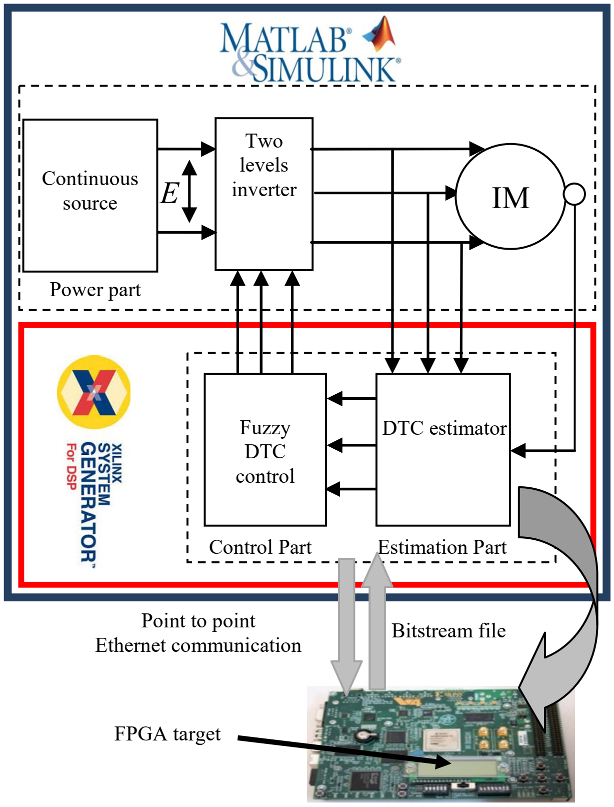
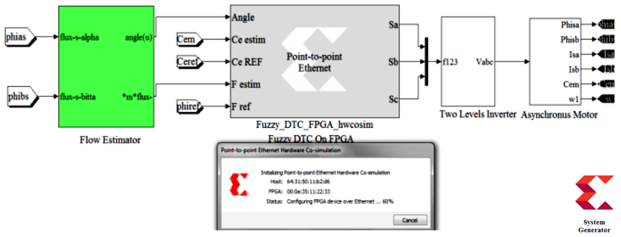


Fig. 20. Hardware in the loop validation of fuzzy DTC controller

Once simulation and timing analysis are done, the procedure of hardware co-simulation in XSG makes a bitstream file from the hardware prototype and a point to point Ethernet block for Hardware-In-the-Loop (HIL) procedure (Fig. 21).



Point-to-point Ethernet Hardware Co-simulation

Initializing Point-to-point Ethernet Hardware Co-simulation
Host: 64:31:50:11:b2:d6
FPGA: 00:0a:35:11:22:33
Status: Configuring FPGA device over Ethernet ... 60%

Fig. 21. Fuzzy DTC HIL point to point Ethernet block

The created block (Fig. 20) substitutes the architecture hardware that was constructed before (fuzzy DTC).

The point-to-point Ethernet blocks are linked to inverter and IM to run a HIL (Fig. 22). In this situation the models of motor and inverter are simulate in MATLAB/Simulink environment, and XSG architectures of Fuzzy DTC are achieved in the ML402 FPGA device. The HIL validation is executed by connecting the target device to PC via an Ethernet cable.

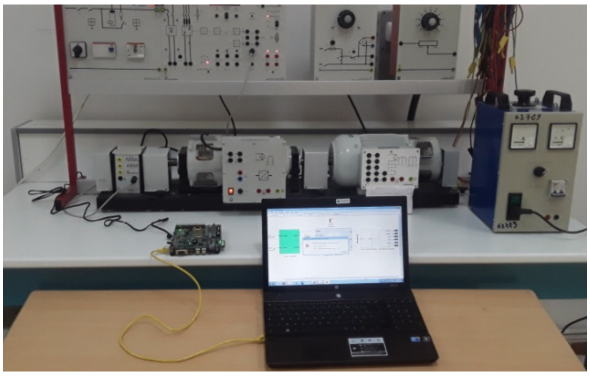


Fig. 22. Fuzzy DTC point to point Ethernet hardware in the loop process

Figure 23 show the waveforms of speed, torque and flux of IM controlled by fuzzy DTC control with co-simulation.

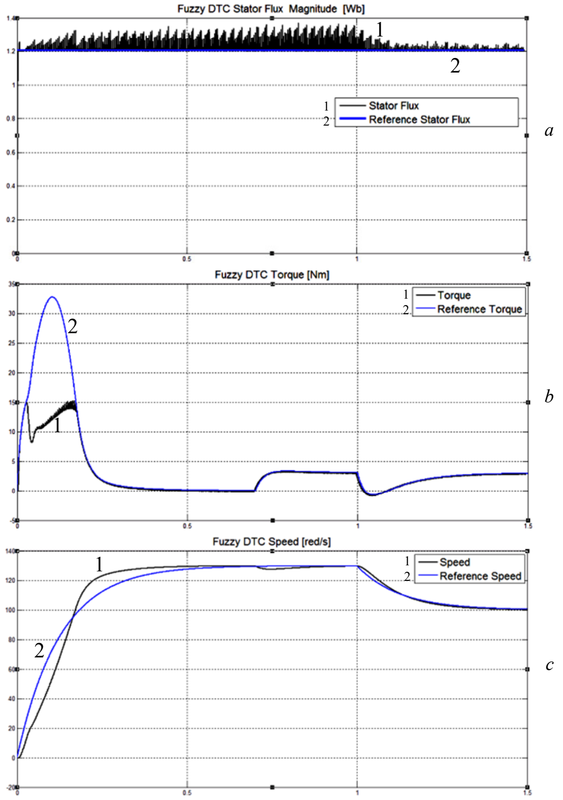


Fig. 23. Behavior of IM flux, torque and speed

The spectral analysis by MATLAB with Powergui FFT analysis tool of electromagnetic torque obtained by conventional DTC (Fig. 24) proposed in [14], the neuronal DTC (Fig. 25) proposed in [7] and by fuzzy DTC (Fig. 26) in steady state shows the existence of harmonics along the spectrum of electromagnetic torque obtained by conventional DTC unlike electromagnetic torque obtained by Fuzzy DTC. Table 5 shows the root mean square (RMS) error and the maximum ripple band of electromagnetic torque and stator flux for conventional DTC, the neuronal DTC and fuzzy DTC approaches.

The results of Table 5 show that DTC based on intelligent techniques considerably reduces the ripples of electromagnetic torque and stator flux compared to conventional DTC. Fuzzy DTC architecture gives the best results in terms of hardware resource consumption and in terms of electromagnetic torque ripple elimination.

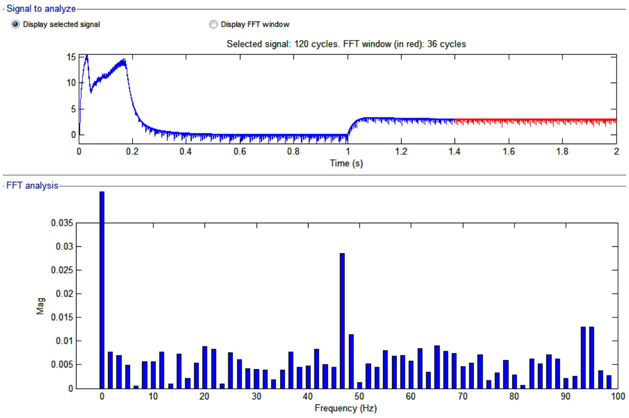


Fig. 24. Spectral analysis of electromagnetic torque obtained by conventional DTC

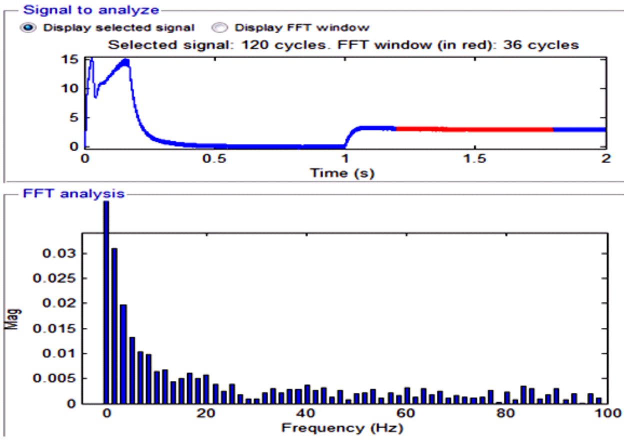


Fig. 25. Spectral analysis of electromagnetic torque obtained by neuronal DTC

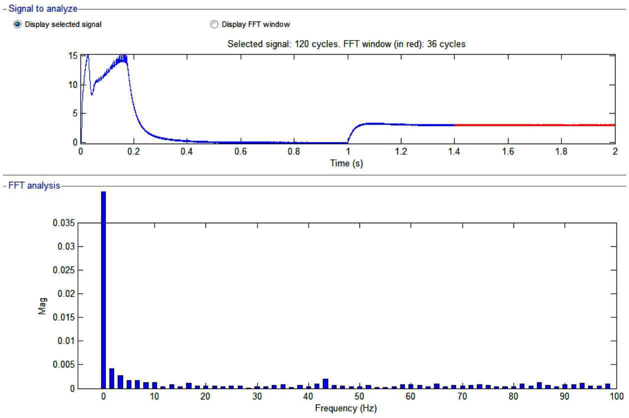


Fig. 26. Spectral analysis of electromagnetic torque obtained by fuzzy DTC

Table 5

Torque and flux ripples

		RMS error	max/min
Electromagnetic torque, N·m	Conventional DTC	0.0367	2.164
	Neuronal DTC	0.0314	0.955
	Fuzzy DTC	0.0096	0.827
Stator flux magnitude, Wb	Conventional DTC	0.0024	0.250
	Neuronal DTC	0.0011	0.090
	Fuzzy DTC	0.0017	0.171

Conclusions.

1. The aim of this work was, first of all, to improve the dynamic performance of the direct torque control applied to induction motor supplied by a voltage inverter by introducing of a fuzzy inference system. Secondly, to materialize the feasibility and to judge the quality of proposed control.

2. In this article, we mainly describe the development, implementation and validation of hardware architecture on field programmable gate array for fuzzy direct torque control of induction motor.

3. The originality of this work has been to combine the performance of artificial intelligence techniques and execution power of programmable logic circuits, for the definition of a control structure achieving the best simplicity / performance and speed / performance ratios.

4. We used unconventional control tools to implement a switching strategy without needing the switching table and hysteresis comparators used in conventional direct torque control.

5. Finally, we believe that the proposed solution improved the dynamic performance of induction motor and greatly reduced the disadvantages of conventional direct torque control such as torque ripples, flux ripples and switching frequency.

Conflict of interest. The authors declare that they have no conflicts of interest.

REFERENCES

1. Takahashi I., Ohmori Y. High-performance direct torque control of an induction motor. *IEEE Transactions on Industry Applications*, 1989, vol. 25, no. 2, pp. 257-264. doi: https://doi.org/10.1109/28.25540.

2. Moussaoui L. Performance enhancement of direct torque control induction motor drive using space vector modulation strategy. *Electrical Engineering & Electromechanics*, 2022, no. 1, pp. 29-37. doi: https://doi.org/10.20998/2074-272X.2022.1.04.

3. Gdaim S., Mtibaa A., Mimouni M.F. Direct Torque Control of Induction Machine based on Intelligent Techniques. *International Journal of Computer Applications*, 2010, vol. 10, no. 8, pp. 29-35. doi: https://doi.org/10.5120/1500-2017.

4. Malyar V.S., Hamola O.Y., Maday V.S., Vasylchyshyn I.I. Mathematical modelling of starting modes of induction motors with squirrel-cage rotor. *Electrical Engineering & Electromechanics*, 2021, no. 2, pp. 9-15. doi: https://doi.org/10.20998/2074-272X.2021.2.02.

5. Djamila C., Yahia M. Direct Torque Control Strategies of Induction Machine: Comparative Studies. *In Direct Torque Control Strategies of Electrical Machines. IntechOpen*. 2021. doi: https://doi.org/10.5772/intechopen.90199.

6. Gdaim S., Mtibaa A., Mimouni M.F. Design and Experimental Implementation of DTC of an Induction Machine Based on Fuzzy Logic Control on FPGA. *IEEE Transactions on Fuzzy Systems*, 2015, vol. 23, no. 3, pp. 644-655. doi: https://doi.org/10.1109/TFUZZ.2014.2321612.

7. Eddine Khodja D., Simard S., Beguenan R. Implementation of optimized approximate sigmoid function on FPGA circuit to use in ANN for control and monitoring. *Control Engineering and Applied Informatics*, 2015, vol. 17, no. 2, pp. 64-72.

8. Da Costa C., Santin C.O. FPGA design approach of digital control of three-phase induction motor. *2017 Brazilian Power Electronics Conference (COBEP)*, 2017, pp. 1-6. doi: https://doi.org/10.1109/COBEP.2017.8257221.

9. Monmasson E., Cirstea M.N. FPGA Design Methodology for Industrial Control Systems – A Review. *IEEE Transactions on Industrial Electronics*, 2007, vol. 54, no. 4, pp. 1824-1842. doi: https://doi.org/10.1109/TIE.2007.898281.

10. Toh C.L., Idris N.R.N., Yatim A.H.M., Muhamad N.D., Elbuluk M. Implementation of a New Torque and Flux Controllers for Direct Torque Control (DTC) of Induction Machine Utilizing Digital Signal Processor (DSP) and Field

Programmable Gate Arrays (FPGA). *IEEE 36th Conference on Power Electronics Specialists*, 2005, pp. 1594-1599. doi: https://doi.org/10.1109/PESC.2005.1581843.

11. Lis J., Kowalski C.T., Orlowska-Kowalska T. Sensorless DTC control of the induction motor using FPGA. *2008 IEEE International Symposium on Industrial Electronics*, 2008, pp. 1914-1919. doi: https://doi.org/10.1109/ISIE.2008.4677287.

12. Sutikno T., Idris N.R.N., Jidin A.Z., Daud M.Z. FPGA based high precision torque and flux estimator of direct torque control drives. *2011 IEEE Applied Power Electronics Colloquium (IAPEC)*, 2011, pp. 122-127. doi: https://doi.org/10.1109/IAPEC.2011.5779871.

13. Monmasson E., Idkhajine L., Cirstea M.N., Bahri I., Tisan A., Naouar M.W. FPGAs in Industrial Control Applications. *IEEE Transactions on Industrial Informatics*, 2011, vol. 7, no. 2, pp. 224-243. doi: https://doi.org/10.1109/TII.2011.2123908.

14. Aib A. FPGA Hardware in the Loop Validation of Torque and Flux Estimators for Direct Torque Control (DTC) of an Induction Motor (IM). *International Journal of Intelligent Engineering and Systems*, 2021, vol. 14, no. 5, pp. 583-594. doi: https://doi.org/10.22266/ijies2021.1031.51.

15. Boukadida S., Gdaim S., Mtibaa A. Hardware Implementation of FTC of Induction Machine on FPGA. *Electronics ETF*, 2017, vol. 20, no. 2, pp. 76-84. doi: https://doi.org/10.7251/ELS1620076B.

16. Zare M.A., Kavasseri R.G., Ababei C. FPGA-based design and implementation of direct torque control for induction machines. *2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14)*, 2014, pp. 1-6. doi: https://doi.org/10.1109/ReConFig.2014.7032520.

17. Lee G.S., Lee D.H., Yoon T.W., Lee K.B., Song J.H., Choy I. Speed and flux estimation for an induction motor using a parameter estimation technique. *International Journal of Control, Automation and Systems*, 2005, vol. 3, no. 1, pp. 79-86.

18. Sandre-Hernandez O., Rangel-Magdaleno J., Morales-Caporal R., Bonilla-Huerta E. HIL simulation of the DTC for a three-level inverter fed a PMSM with neutral-point balancing control based on FPGA. *Electrical Engineering*, 2018, vol. 100, no. 3, pp. 1441-1454. doi: https://doi.org/10.1007/s00202-017-0597-0.

19. Lotfi E., Elharoussi M., Abdelmounim E. VHDL design and FPGA implementation of direct torque control for induction machines. *Bulletin of Electrical Engineering and Informatics*, 2021, vol. 10, no. 3, pp. 1220-1231. doi: https://doi.org/10.11591/eei.v10i3.2345.

Received 25.07.2022
Accepted 15.11.2022
Published 06.05.2023

Abdelghani Aib [1], *Doctor of Electrotechnical,*
Djalal Eddine Khodja[2], *Doctor of Electrotechnical, Professor,*
Salim Chakroune [1], *Doctor of Electrotechnical, Professor,*
[1] Research Laboratory on the Electrical Engineering,
Faculty of Technology,
University of M'Sila, BP 166, Ichbilia 28000, Algeria,
e-mail: abdelghani.aib@univ-msila.dz,
salim.chakroun@univ-msila.dz (Corresponding Author).
[2] Signals & Systems Lab,
Institute of Electrical and Electronic Engineering,
Boumerdes, 35000, Algeria,
e-mail: djalaleddine.khodja@univ-msila.dz

How to cite this article:
Aib A., Khodja D.E., Chakroune S. Field programmable gate array hardware in the loop validation of fuzzy direct torque control for induction machine drive. *Electrical Engineering & Electromechanics*, 2023, no. 3, pp. 28-35. doi: https://doi.org/10.20998/2074-272X.2023.3.04