Received: June 14, 2021. Revised: August 11, 2021.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

FPGA Hardware in the Loop Validation of Torque and Flux Estimators for Direct Torque Control (DTC) of an Induction Motor (IM)

Abdelghani Aib[1]* **Djalal Eddine Khodja[1,2]** **Loutfi Benyettou[1]** **Salim Chakroune[1]**

[1]*Research Laboratory on the Electrical Engineering, Faculty of Technology, University of M'Sila, Algeria*
[2]*Research Laboratory on the Electrification of Industrial Enterprises, Boumerdes University, Algeria*
* Corresponding author's Email: abdelghani.aib@ univ-msila.dz

Abstract: This article provides a hardware implementation of torque and flux estimators for direct torque control (DTC) of an induction motor (IM) on the Field Programmable Gate Array (FPGA). Due to its high processing frequency, the FPGA circuit presents an alternative strategy for achieving a high performance DTC implementation. This cannot be achieved by Digital Signal Processing (DSP) or Microcontroller application. Optimized hardware architecture is also proposed for implementation of a complete estimator with explicit determination of all the quantities such as the flux phase, the flux amplitude and the torque amplitude. This work presents a precise hardware estimator of the torque and the flux, it developed with negligible estimation error (0.0002 [Wb] RMS estimate error for the stator flux and 0.0005 [Nm] RMS estimate error the torque), in addition to that this estimator is implemented on the Virtex-4 FPGA board with minimal resources(less 5% of slices registers, 5% of DSP48Es and around 17% of LUTS). This estimator is developed and synthesized independently of the command and can be reused for any task which requires the estimation of these quantities. After this the estimator was used to complete a DTC control in a hardware co-simulation procedure with the Xilinx system generator and MATLAB/SIMULINK tools on the Xilinx ML402 development kit.

Keywords: FPGA, Hardware co-simulation, Xilinx generator system, Induction motor control, Flux and torque estimation.

Nomenclature

DTC	Direct Torque Control	$\emptyset_{s\alpha}$, $\emptyset_{s\beta}$	Stator flux in α-β reference frame
SVM	Space Vector Modulated	$\|\|\emptyset_s\|\|$	Stator flux magnitude
IM	Induction Motor	$\theta_{\emptyset_s}$	Stator flux Angle
FPGA	Field Programmable Gate Array	T_e	Electromagnetic Torque
DSP	Digital Signal Processing	I_{sa}, I_{sb}	Stator currents in the abc reference frame
XSG	Xilinx System Generator	T_{eref}	Reference Electromagnetic Torque
VHDL	VHSIC Hardware Description Language	$\emptyset_{sref}$	Reference Stator Flux
CORDIC	Coordinate Rotation Digital Computer	N	Sectors number
Zn	CORDIC scale factor	$E_\emptyset$	Flux error
RMS	Root Mean Square	E_{Te}	Torque error
FFT	Fast Fourier Transform	E	Constant voltage
LUT	Look Up Tables	S_a, S_b and S_c	Boolean switching commands
IOB	Input Output Block	ω	Speed
$V_{s\alpha}$, $V_{s\beta}$	Stator voltages in α-β reference frame		
$I_{s\alpha}$, $I_{s\beta}$	Stator currents in α-β reference frame		

 DOI: 10.22266/ijies2021.1031.51

ω_{ref}	Reference Speed
R_s	Stator resistance
Ts	Sampling Time
P	Machine pair pole number
$\bar{V}_s$	Boolean switching commands vector

1. Introduction

Considering the advantages of the asynchronous machine in terms of robustness and lower cost, it is the machine widely used in electric drives. Advances in power electronics and microelectronics have made it a formidable competitor in the fields of variable speed and rapid torque control [1].

Direct torque control (DTC) was first introduced by Dopenbrock and Takahashi [2, 3]. It is one of the methods used in variable frequency drives such as induction motor drives. Its fame was derived from the simplicity of its structure, its high efficiency and low cost [4-6].

An efficient DTC of the induction motor drive systems involves fast computational units. Digital Signal Processing (DSP) and microprocessors are frequently used in such applications [7-9]. But these devices have a limited processing speed because of the serial calculations; this affects its performance especially in real time application. The use of FPGA circuits offers an appropriate solution for fast calculations. A strategy for FPGA programming and multilevel inverter control was presented in [10, 11]. In [12-14] an FPGA based DTC implementation was developed. An FPGA based torque and flux estimator for DTC control was implemented in [15]. In [16, 17] an FPGA Hardware Co-Simulation process for induction motor control was presented.

However, the most difficult parts of DTC control to implement are the torque and flux estimators [18]. In fact, complex numerical calculations are called upon during implementation, such as binary multiplication, the Arctan function and also the square root [19-21].

The number of hardware resources in a FPGA circuit being limited, some efforts must be made to minimize these resources through a simple design methodology proposed in this article, and which will be applied to the proposed material architectures. This methodology is based on the illustration of a compromise that performs an optimization of hardware architecture while preserving the good performances of electrical drive system consisting of a voltage inverter and an asynchronous machine.

This paper presents an efficient Hardware architecture implemented with VHDL and XSG was used for the implementation of the proposed estimator and the DTC command on FPGA. We used an optimized version of the CORDIC algorithm implemented in VHDL for the most expensive tasks in terms of computational skills and resources consumption. After the validation step our estimator was used to achieve a DTC control in a FPGA hardware co-simulation process. The obtained results show the efficiency of the proposed architecture in terms of the resource consumption, the accuracy of the estimation and the dynamic performance of the induction machine.

We have begun this work with a general introduction presenting the context of the different methods of digital implementation on FPGA target of DTC control and torque and flux estimators, then we will study in section 2 the principle of the DTC control and the mathematical equations of this estimators and this command strategy.

In section 3, we presented and detailed the design and the proposed architecture for our estimator and the DTC control block by block, which we will use later to build a hardware co-simulation. In section 4, a Simulink/XSG Hardware-in-the-loop platform is given for the validation of the proposed estimator in the DTC control loop, the results of the proposed architecture are presented and discussed, the general conclusion and the future works are presented at the end of this paper.

2. Direct torque control principle

The direct torque control of the asynchronous machine is based on the application of the different voltage vectors of the inverter, which are controlled in two variables are the stator flux and the electromagnetic torque by hysteresis regulators. The power converter is a conventional two-level voltage inverter. These values are directly estimated from the stator voltages determined with the constant voltage E and the Boolean switching commands (Sa, Sb, Sc). The estimated values of the torque and of the stator flux are compared respectively with their estimated reference values and; the comparison results are the inputs to the hysteresis cycle comparators. The controlled electromagnetic torque is provided by the PI speed controller (see Fig. 1).

Moreover, several parameters must be determined in order to estimate the stator flux and the electromagnetic torque. Their models are adapted to the needs of controlled training. First, the stator currents a-b-c are transformed into α-β coordinates, and the stator voltages in the α-β frame of reference are determined according to the switching state (Sa, Sb and Sc) produced by the switching table [21].

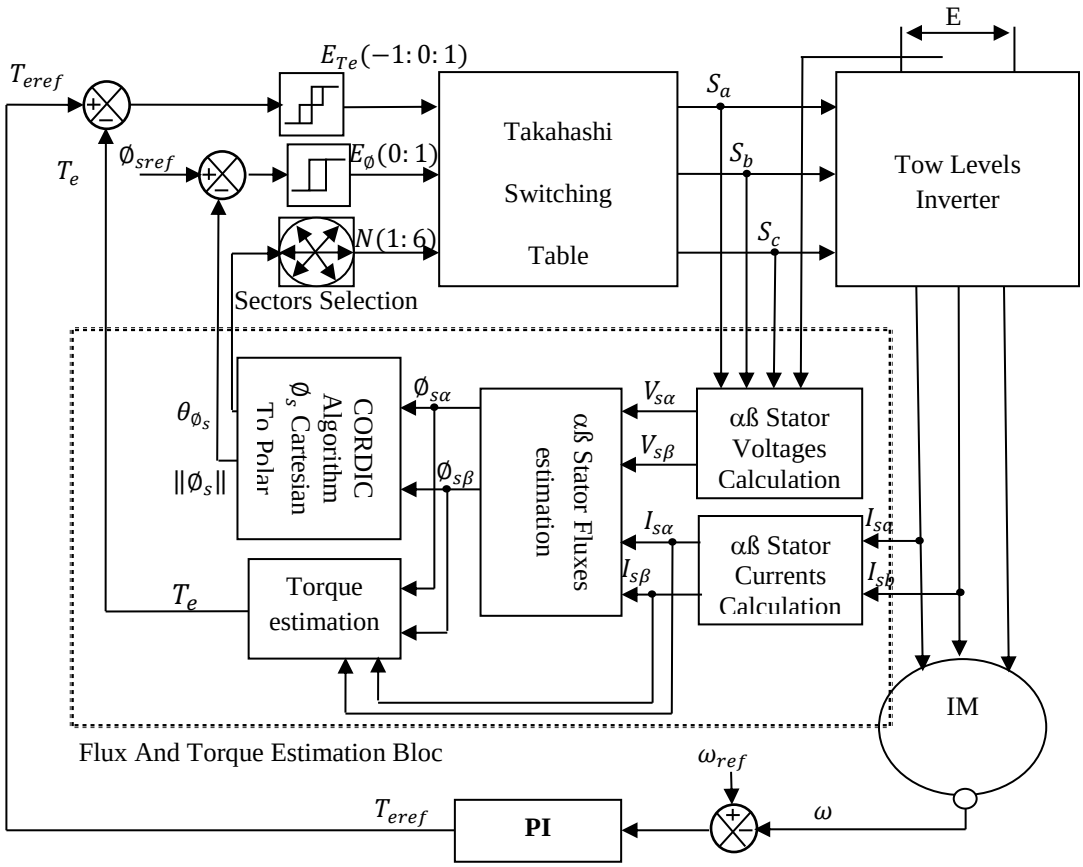


Figure. 1 Functional schems of DTC control

Using the stator current and voltage, the stator flux can be estimated in coordinates (α-β) as follows [21]:

$$\emptyset_{s\alpha} = \int (V_{s\alpha} - R_s i_{s\alpha}) dt \quad (1)$$

$$\emptyset_{s\beta} = \int \left(V_{s\beta} - R_s i_{s\beta}\right) dt \quad (2)$$

To perform the integral function, the Euler approach is used with a sampling time (Ts), the system (1) and (2) become as follow [21]:

$$\emptyset_{s\alpha} = \emptyset_{s\alpha Old} + T_s (V_{s\alpha} - R_s i_{s\alpha}) \quad (3)$$

$$\emptyset_{s\beta} = \emptyset_{s\beta Old} + T_s \left(V_{s\beta} - R_s i_{s\beta}\right) \quad (4)$$

Finally, The electromagnetic torque (Te) can be estimated by [21]:

$$T_e = \frac{3}{2} P \left(\emptyset_{s\alpha} i_{s\beta} - \emptyset_{s\beta} i_{s\alpha}\right) \quad (5)$$

For the stator flux magnitude ($\|\emptyset_s\|$) and Angle ($\theta_{\emptyset_s}$) can be obtained by Cartesian to Polar Coordinates Transformation. Unfortunately, the standard mathematics transformation based on the formulas: $\|\emptyset_s\| = \sqrt{{\emptyset_{s\alpha}}^2 + {\emptyset_{s\beta}}^2}$ and $\theta_{\emptyset_s} = ArcTan(\frac{\emptyset_{s\beta}}{\|\emptyset_s\|}, \frac{\emptyset_{s\alpha}}{\|\emptyset_s\|})$ is not always implantable on hardware and often requires a lot of device resources because of excessive resources functions such as the binary multiplication and division, the Square Root and the ArcTan function. To overcome this constraint, our estimator uses a version of CORDIC (coordinate rotational digital computer) algorithm [22] to calculate the stator flux magnitude and angle, using simply implementable operators on FPGA like addition, subtraction and binary digits shift [23].

T. Sutikno et al. [18] used two-binary multiplier and 62 bits no restoring square root [24] for calculating flux magnitude. And [20] used two-binary multiplier, Square Root generalized CORDIC bloc and ArcTan generalized CORDIC bloc to estimate flux magnitude and angle. Two-binary multiplier and Square Root generalized CORDIC bloc was used to estimate flux magnitude in [21]. Our solution uses only reduced CORDIC bloc and one multiplier to estimate simultaneously the flux magnitude and angle.

3. Direct torque control conception using XSG

3.1 Implementation of flux and torque estimator

In this part, flux and torque estimators are designed based on FPGA circuits to improve DTC induction motor drives.

This allows precise calculations and very fast response times. To do this, a discrete integration operation calculation was performed for the magnitude of the stator flux using Euler's recursive approach, and for the angle and the magnitude of the stator flux the CORDIC Cartesian Polar Coordinate algorithm was used. In addition, to facilitate the description of the material, the models of the estimation diagram are simplified and organized under different blocks (Fig. 2).

In fact, for the verification and the development of the algorithms of the estimators in the FPGA circuit we will use the modeling tool known as XSG developed by Xilinx intended for MATLAB / SIMULINK.

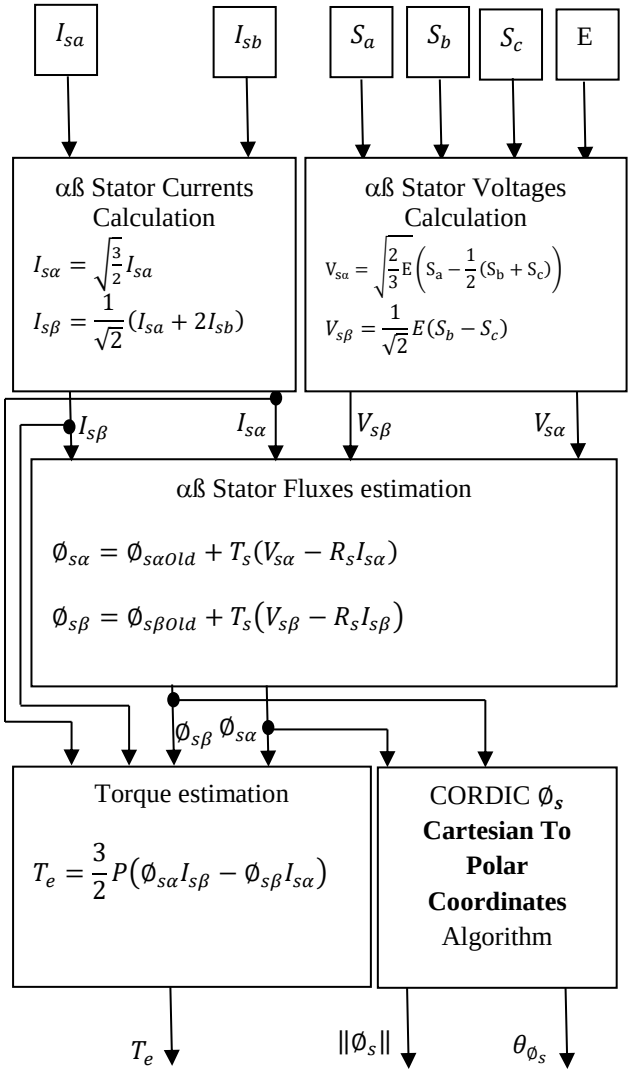


Figure. 2 Flow hardware of torque and flux estimator

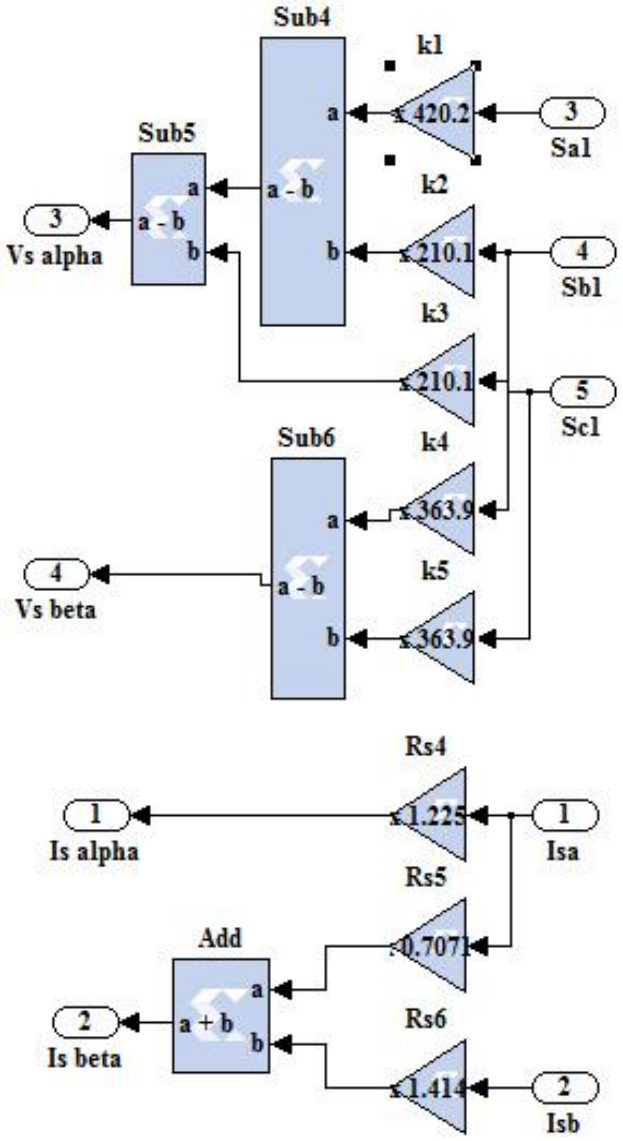


Figure. 3 FPGA α-β stator current and voltage calculator implementation

The main role of the XSG tool is to transform the DTC model into hardware format for implementation on FPGA. Then it generates VHDL code without any difficult programming. Indeed, the implementation time of this algorithm turns out to be reduced.

3.1.1. Implementation of α-β stator current and voltage calculator

XSG design of the (α-β) stator voltage and current calculator are shown in Fig. 3.

3.1.2. Implementation of αß stator fluxes estimation

The Fig. 4 shows the hardware design for the estimator of α-β stator flux using the System Generator.

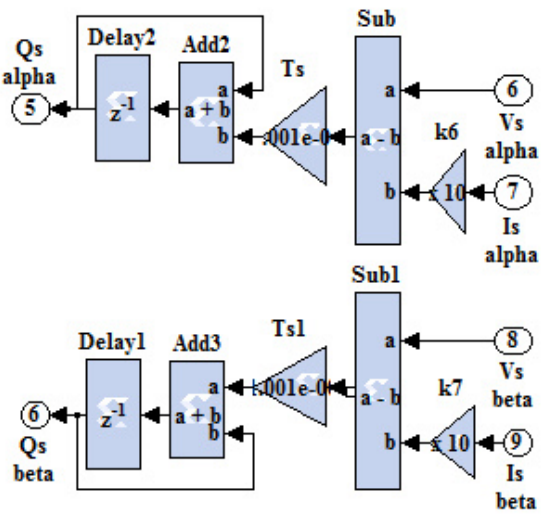


Figure. 4 Hardware design for estimator of α-β stator flux using System Generator

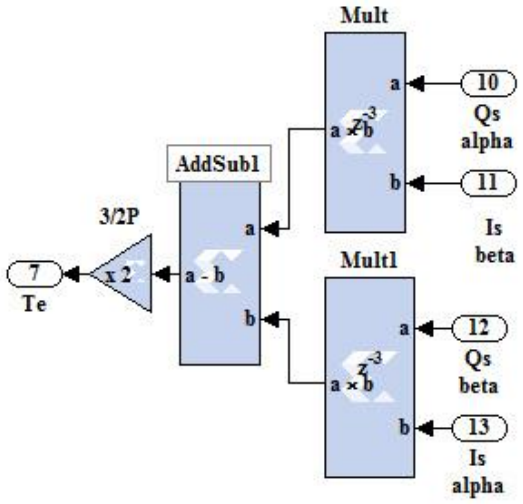


Figure. 5 Hardware design for the torque estimator using System Generator

3.1.3. Implementation of torque estimator

Fig. 5 shows the torque estimator design of proposed DTC using System Generator.

3.1.4. Implementation of CORDIC $\emptyset_s$ cartesian to polar coordinates transformation

The CORDIC algorithm is designed to perform a vector rotation of the vector (X, Y) through an angle θ, resulting in a new vector (X ', Y') [23].

$$\begin{cases} x' = (x\,cos(\theta) - y\,sin(\theta)) \\ y' = (y\,cos(\theta) + x\,sin(\theta)) \\ \theta' = \theta \end{cases} \quad (6)$$

The role of the CORDIC algorithm is to make a vector rotation by successive and smaller sequences of rotations, each angle atan (2-i), called micro-rotations. Eq. (7) shows the expression of the ith iteration where i is the iteration index from 0 to n [11].

$$\begin{cases} x_{i+1} = x_i - \alpha_i y_i\, 2^{-i} \\ y_{i+1} = y_i + \alpha_i x_i\, 2^{-i} \\ \theta_{i+1} = \theta_i + \alpha_i atan(2^{-i}) \\ \alpha_i = (+1\ or\ -1) \end{cases} \quad (7)$$

Where α_i is the direction of rotation.

In fact, a micro-rotation is expressed as a simple operation of shift and addition / subtraction. Eq. (8) shows the vector rotation expression for the nth iteration [11].

$$\begin{cases} x_n = \prod_{i=0}^{n-1} cos(atan(2^{-i}))(x_i - \alpha_i y_i\, 2^{-i}) \\ y_n = \prod_{i=0}^{n-1} cos(atan(2^{-i}))(y_i + \alpha_i x_i\, 2^{-i}) \\ \theta_n = \sum_{i=0}^{n-1} \alpha_i atan(2^{-i}) \\ \alpha_i = (+1\ or\ -1) \end{cases} \quad (8)$$

The outputs of Eqs. (7) and (8) of the CORDIC algorithm are equivalent to a vector rotation or vector translation scaled by a constant Zn. In this case the constant Zn is known as the CORDIC scale factor. The Taylor series expansion of $cos(atan(2^{-i}))$ is $(1 + 2^{-2i})^{-1/2}$. Hence, the constant Zn can be expressed as:

$$Zn = \prod_{i=0}^{n-1} \frac{1}{(1+2^{-2i})^{-1/2}} \quad (9).$$

The CORDIC scale factor of Zn depends only on the number of iterations 'N'. In other words, Only the functional Rotation and Translation configurations: Rectangular to Polar and Polar to Rectangular are affected by the CORDIC scale factor. However, once these functional configurations are selected, the CORDIC core offers the possibility of multiplying by Zn to cancel the scale factor. For n=10, Zn=0.6073.

The previous CORDIC algorithm is implemented in hardware by VHDL architecture:

```
Entity CORDIC is
Port ( Qsd: in STD_LOGIC_VECTOR (15 down to 0);
    Qsq: in STD_LOGIC_VECTOR (15 down to 0);
    Magnitude: out STD_LOGIC_VECTOR (15 down to 0);
    Angle_Q: out STD_LOGIC_VECTOR (15 down to 0));
End CORDIC;
```

The VHDL CORDIC code is implemented in Black Box with XSG:

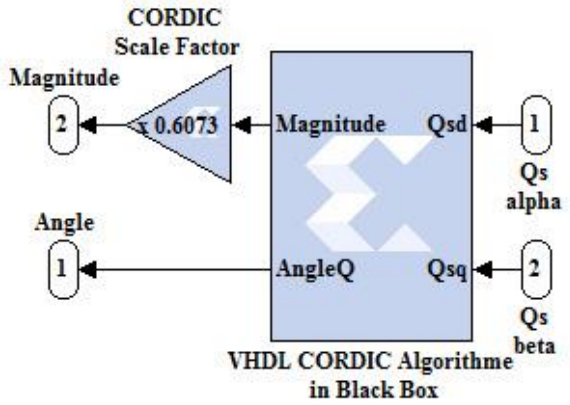


Figure 6. CORDIC algorithm with XSG

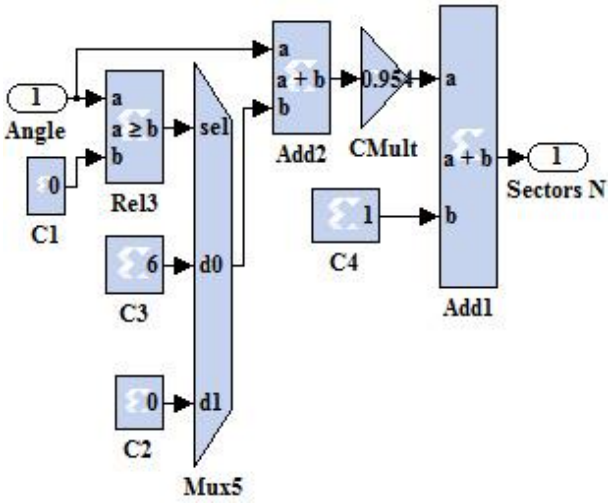


Figure. 7 XSG of sectors selection implementation

3.2 Implementation of sectors selection

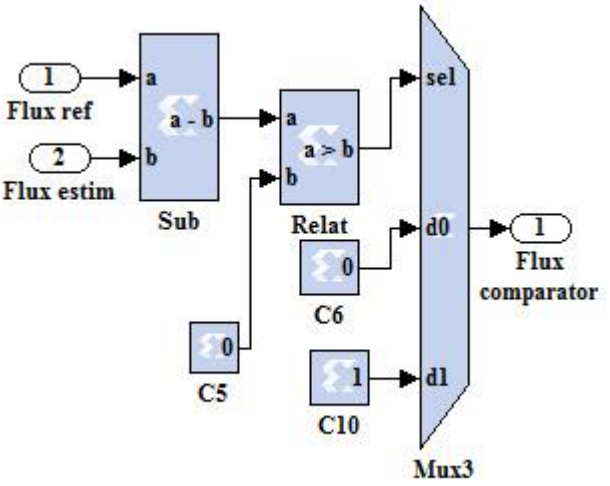
The block in Fig. 7 determines in which sector number the stator flux vector is located at a given sampling time. In fact, the coordinate plane α-β is subdivided into 6 sectors. This is done by comparing the flow angle with the boundaries of each sector. For this, we use two comparators, two adders and a multiplexer in XSG.

3.3 Implementation of hysteresis controller

Fig. 8 shows the XSG implementation of the three and two level hysteresis comparators of flux and torque respectively. As for the estimated flux and torque values are compared with the reference values (torque and flux control values).

3.4 Implementation of takahashi switching table

The switching signals of the IGBTs of the inverter are determined according to the comparison results of



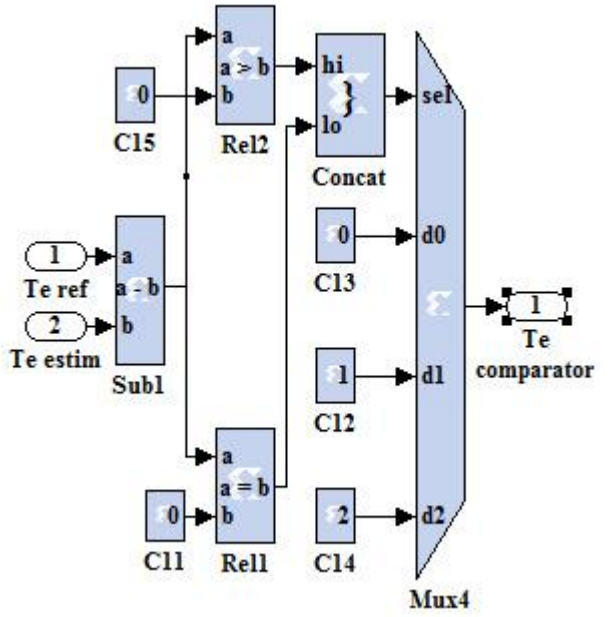
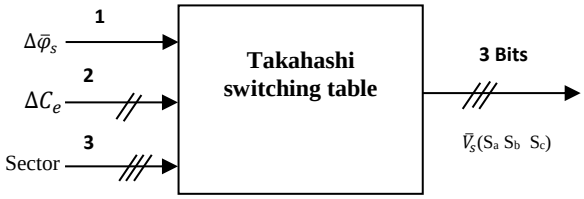


Figure. 8 Hysteresis comparators XSG implementation

the hysteresis regulators from the switching table shown in Table 1. The table consists of null and active vectors. The zero vectors are (0,0,0) and(1,1,1) which stop the field vector, thus reducing the torque. While, the other six active vectors advance the field forward and resulting in increased torque [25].

Table 1. Switching table for DTC

		$V_i = (S_a\ S_b\ S_c)$					
$\Delta\bar{\varphi}_s, \Delta T_e, N$		N_1	N_2	N_3	N_4	N_5	N_6
$\Delta\bar{\varphi}_s$=1	ΔT_e=1	(110)	(010)	(011)	(001)	(101)	(100)
	ΔT_e=0	(111)	(000)	(111)	(000)	(111)	(000)
	ΔT_e=-1	(101)	(100)	(110)	(010)	(011)	(001)
$\Delta\bar{\varphi}_s$=	ΔT_e=1	(010)	(011)	(001)	(101)	(100)	(110)
	ΔT_e=0	(000)	(111)	(000)	(111)	(000)	(111)
	ΔT_e=-1	(001)	(101)	(100)	(110)	(010)	(011)




```
entity DTC is
    Port ( Selec : in  STD_LOGIC_VECTOR (2 downto 0);
           Cflex : in  STD_LOGIC;
           Ccpl : in  STD_LOGIC_VECTOR (1 downto 0);
           S135 : out STD_LOGIC_VECTOR (2 downto 0) );
end DTC;
```

Figure. 9 Takahashi switching table VHDL architecture

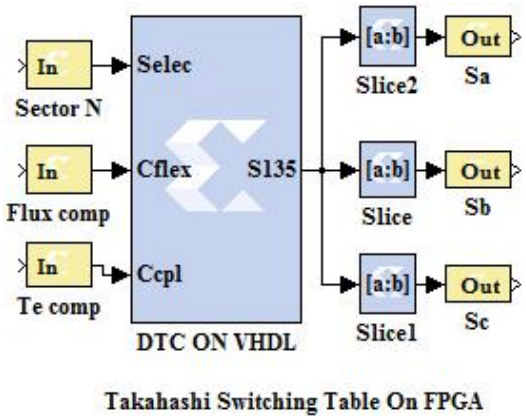


Figure. 10 Takahashi switching table with XSG

The Takahashi Switching Table is implemented by VHDL hardware architecture. The VHDL hardware description language allows us to define the inputs/outputs of our architecture. An input bit for the flux error $\Delta\bar{\varphi}_s$(0 or 1), two input bits for the torque error ΔT_e(-1,0 or 1), three input bits for the Sectors selection (N_1, N_2, N_3, N_4, N_5 or N_6) and three output bits for the $\bar{V}_i$(S_a, S_b and S_c). The Fig. 9 presents the inputs/outputs of the proposed architecture and the corresponding VHDL code.

The VHDL architecture of Takahashi Switching Table is implemented by an assignment in the competitive mode:

$$\bar{V}_{Sortie} <= \bar{V}_1\ when\ Condition1$$
$$else\ \bar{V}_2\ when\ Condition2$$
$$else\ \bar{V}_3\ when\ Condition3$$
$$else\ \bar{V}_4\ when\ Condition4$$
$$else\ \bar{V}_5\ when\ Condition5$$
$$else\ \bar{V}_6\ when\ Condition6$$
$$else\ \bar{V}_7\ when\ Condition7$$
$$else\ \bar{V}_0$$

The output signal vector $\bar{V}_s = \begin{pmatrix} Sa \\ Sb \\ Sc \end{pmatrix}$ divides by slice to generate the three signals (S_a, S_b and S_c) of the inverter control. A slice extract a given range of bits from each input sample and presents it at the output.

4. Simulation, results and discussions

4.1 Matlab/Simulink and XSG/Xilinx simulations

The implementation of the proposed architectures has been carried out using the XSG blocks available on the Simulink library. Once a system designed gives the desired simulation results, the bitstream file and the Hardware-In-the-Loop block can be generated by system generator. The parameters of the induction motor used in the co-simulation phase are presented in Table 2. The following figure shows the DTC simulation blocks.

Table 2. Parameters of induction motor

Parameter	Value
R_S	10 Ω
R_r	6.3 Ω
L_S	0.4642 H
L_r	0.4612 H
L_m	0.4212 H
J	0.02 kg.m^2
P	2

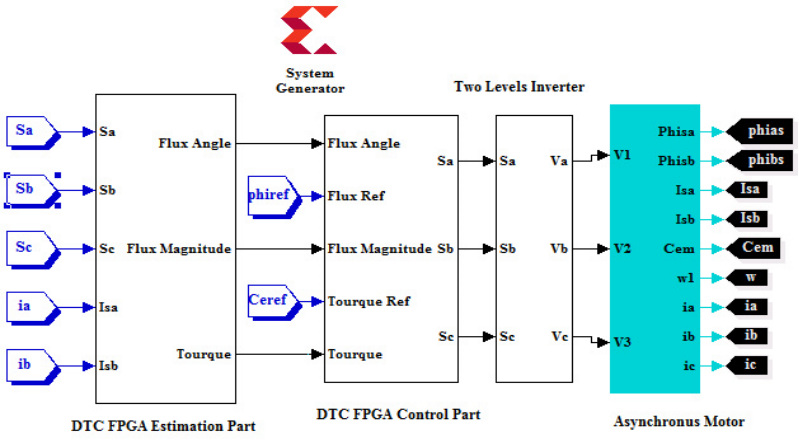


Figure. 11 Proposed DTC XSG/Simulink simulation blocks

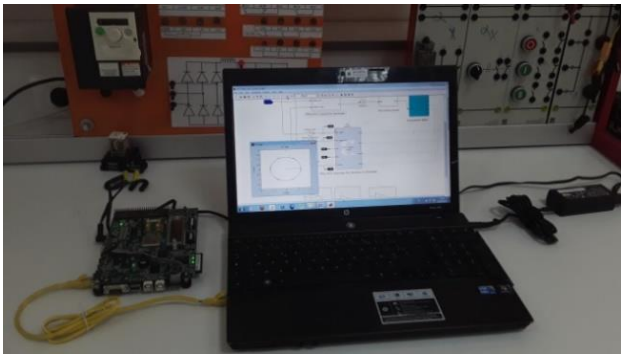


Figure. 12 Test bench for validation by co-simulation hardware

4.2 FPGA hardware co-simulation

4.2.1. Creating point to point Ethernet block for the hardware-in-the-loop

Once the simulation and temporal analysis are performed, the XSG hardware co-simulation process generates a bitstream file and a point-to-point Ethernet block for the Hardware-in-the-loop (HIL) process. The generated block (Fig 13) replaces the hardware design already built (DTC estimation part and DTC control part).

The point to point Ethernet blocks (estimation part and control part) are connected to the inverter to run a HIL. In this situation the models of motor and inverter are simulated in Matlab/Simulink environment, and XSG architectures of DTC Estimation Part or DTC Control part are achieved in the ML402 FPGA device. The validation of hardware-in-the-loop is executed by connecting the target device to the PC with an Ethernet cable.

Fig.12 shows the test bench for the validation of the architecture proposed by hardware Co-Simulation.

4.2.2. Synthesis results

In this step, the VHDL codes are synthesized in order to convert them into XSG blocks. The resource usage of the DTC implementation on the FPGA is shown in Table 3. The latter presents the necessary information of Input Output blocks number, Slices Registers, Slice LUTs (Look Up Tables) and number of DSP.

The proposed architecture significantly reduced the hardware resources of the DTC FPGA implementation compared to the presented in [20], [25].

In this work, the maximum combinatorial path delay: 4.317ns (232 MHz). While in articles [25] the maximum clock frequency is 54 MHz using dSPACE (digital signal processing and control engineering). In [9] the sampling time is equal to 50 µs. However, the execution time is too long compared to the FPGA.

4.2.3. High level simulation of the flux and torque estimator

The proposed flux and torque estimator architecture is validated by co-simulation process in Matlab/Simulink environment with XSG tools and ML402 device.

The estimated stator flux in the plan α-β is shown in Fig. 14.

In the Fig. 15, the calculate torque, flux angle and flux magnitude are presented for the induction motor start-up phase.

Table 4 shows the root mean square error and the maximum error between the theorecal valus and the valus estimated by our estimator of the electromagnetic torque, stator flux magnitude and stator flux angle.

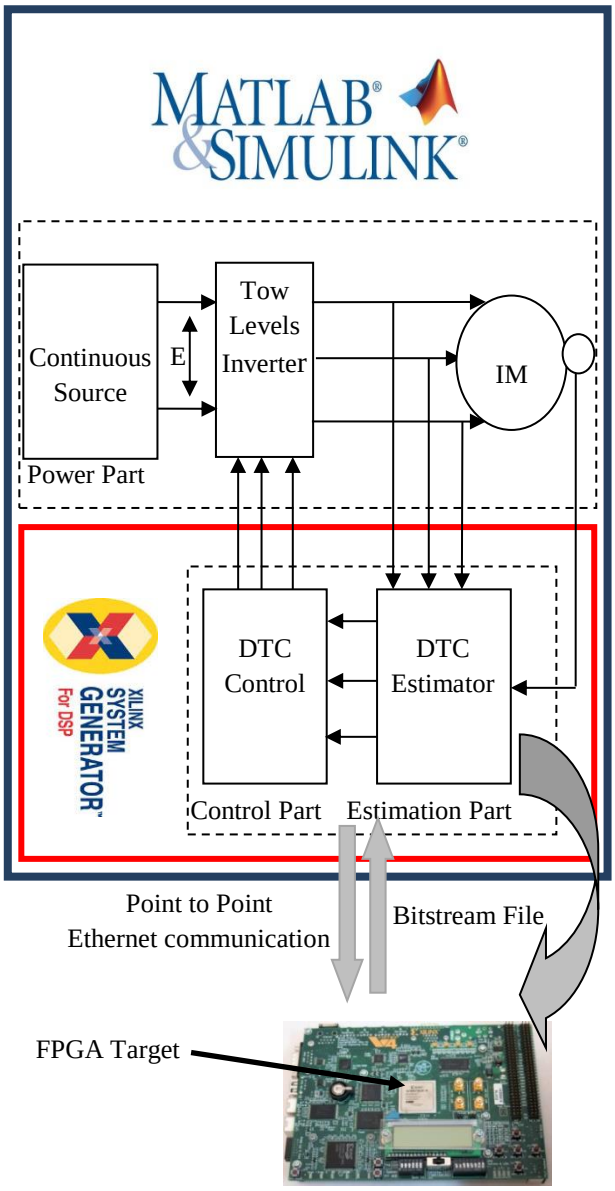


Figure. 13 Proposed DTC hardware co-simulation

Table 3. Used resources

Target Device: ML402 Virtex-4 xc4vsx35-10ff668					
Logic Utilization	Estimation	Control	Total	Available	Utilization
Number of bonded IOBs	58	58		448	
Number of slices registers	1,471	1,438	2,909	30,720	09%
Number of slice LUTS	5,409	2,178	7,587	30,720	25%
Number of DSP48Es	10	00	10	192	05%

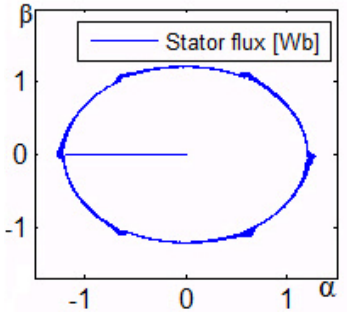


Figure. 14 The stator flux in the plan α-β

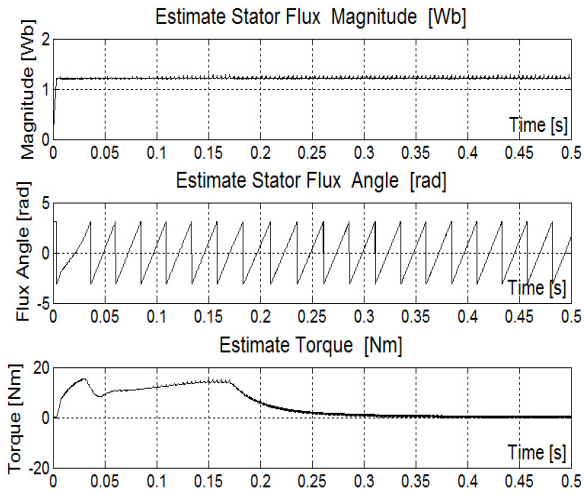


Figure. 15 The estimate flux and torque in start-up phase

Table 4. Comparison of the estimate and theoretical values

	Stator flux angle [rad]	Stator flux magnitude [Wb]	Electromagnetic torque [Nm]
RMS error	0.01	0.0002	0.0005
MAX error	0.03	0.02	0.04

4.2.4. High level simulation of the DTC control performances

The response waveforms of the induction motor speed and torque versus differ reference speed and resistant torqueses are shown in Fig. 16.

The evolution of speed, phase current and flux are presented in Fig. 17.

The results obtained during the hardware co-simulation phase proved that the implementation was successful. Nevertheless, there is the nourishment of certain fluctuations at the levels of the curves of parameters such as current, torque and flux. This is due to the attempt to choose the optimal size of the data in order to achieve an accurate calculation and minimum internal resources of the FPGA circuit.

Indeed, it is necessary to reduce the sampling time in order to reduce the torque ripples.

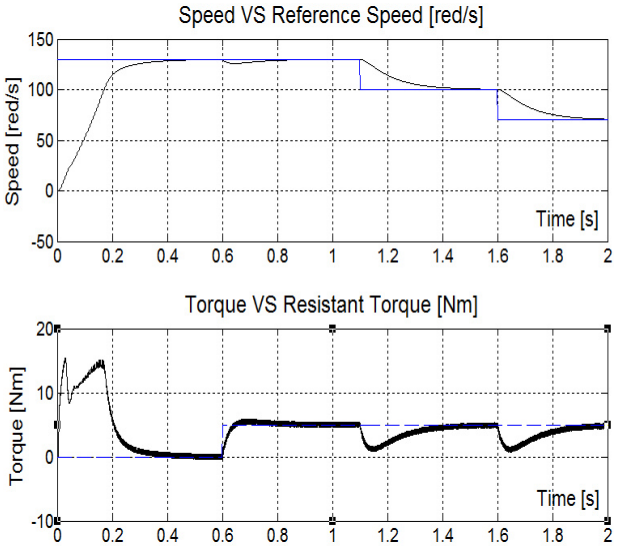


Figure. 16 Behavior of induction motor speed and torque

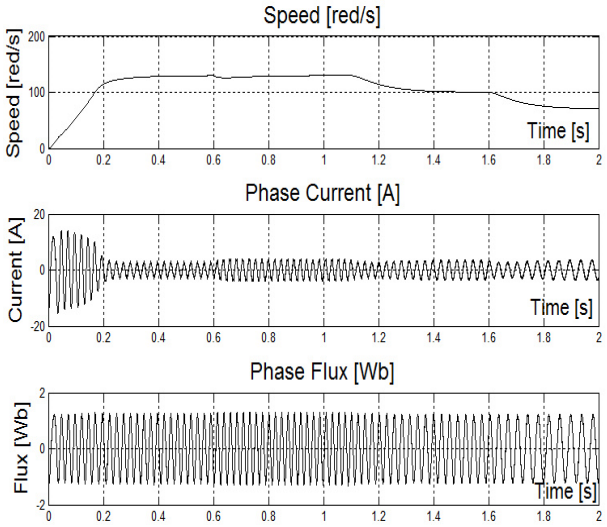


Figure. 17 Behavior of induction motor speed, phase current and phase flux

Furthermore, the fluctuation of the estimated torque and flux does not matter on the dynamic responses of the motor; even the speed has not been affected.

The following figure shows the spectral analysis of the speed curve obtained by "Powergui FFT Analysis Tool" from Matlab.

The spectral analysis by FFT of the speed shows that the ripple of the torque and of the flux generated by the estimation errors does not influence the dynamic response of the speed.

The main objective of this work is the reduction of the hardware architecture for an implementation on FPGA of a flux and torque estimator for the DTC control using optimized functions coded in VHDL, while keeping the quality of dynamic performance of the asynchronous machine. Note that the reduction in the size and the hardware resources of the proposed

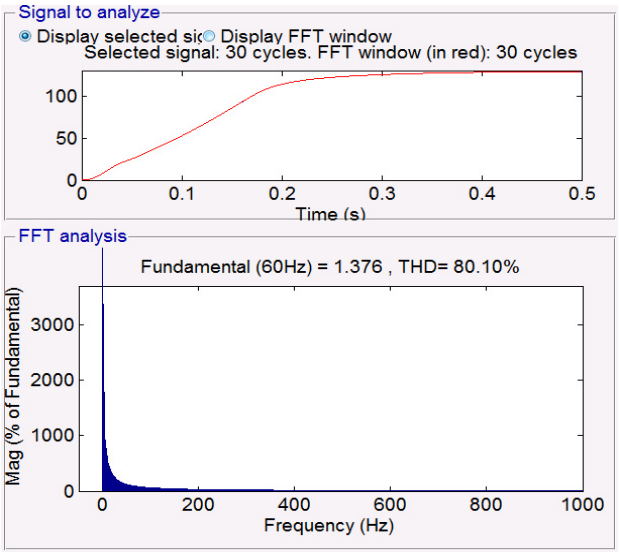


Figure. 18 FFT analysis of speed response

Table 5. Comparison of the resources consumption

Logic Utilization	References			
	[25]	[26]	[27]	Proposed
FPGA device family	Xilinx Virtex-5	Altera DE-115	Altera CYCLONE II	Xilinx Virtex-4
Embedded multiplier 9-bit elements	/	80	57	/
Total logic elements	2.502	6.931	3.256	**2.909**
Total combinational functions	21.758	6.491	2.549	**7.587**

Table 6. Comparison of the estimators performances

References	[21]	[24]	[25]	[26]	[27]	Proposed
Estimate flux magnitude	Yes	yes	yes	Yes	yes	Yes
Estimate flux angle	No	No	No	Yes	no	Yes
Estimate Electromagnetic torque	Yes	yes	yes	Yes	yes	Yes
Optimized estimator	No	yes	No	No	yes	Yes
Reusable estimator	No	No	No	No	no	Yes

architecture implies the minimization of the execution time. Consequently, it implies minimizing consumption and the cost of designing such an order. The Table 5 presents the hardware resources consumed in this architecture compared to previous work in the same research axis.

The Table 5 shows that the proposed architecture minimizes the use of resources. In [27] although the proposed architecture gives satisfactory results in terms of consumption of certain resources such as the logical elements and the combinatorial function but for certain other resources such as the number of pins and the multipliers, the consumption is very high.

In [24] an excellent idea was proposed for the computation of the square root function by a recursive procedure minimizing the resources to compute the amplitude of the stator flux. The implemented estimator only calculates the flux and torque amplitudes but not the stator flux angle. It uses a judge for the identification of sectors without calculating the explicit value of the stator flux angle. This is also applied to all previous work. This makes the flux and torque estimators in this work only applicable for a standard 6 sector DTC control. If we want to implement these estimators for a (12 or 24) sector DTC or an SVM-DTC we have to change or reimplement the estimator.

The Table 6 shows the performances of the proposed estimators compared to the estimators cited in the previous works.

The work presented in this article proposes a universal estimator of the torque and the flux which we can reuse for any IM controls requires these estimated quantities without significant modifications.

5. Conclusion

The aim of this work in the first place is to develop a flux and torque estimator for the direct control of the torque applied to an induction motor.

In a second place is to materialize the feasibility and to judge the quality of the proposed estimator.

In this article, we mainly describe the development, the implementation and the validation of hardware architecture on FPGA for an estimator of the flux and the torque applied to a DTC control of an induction motor.

The originality of this work is the hardware implementation on FPGA of a universal torque and flux estimator capable of explicitly calculating all the quantities necessary for the conventional DTC command and these commands derived from stator currents and trigger commands inverter, while minimizing hardware resources in the proposed architecture using also optimized functions implemented in VHDL.

The proposed estimator developed with negligible estimation error (RMS error of estimate Stator flux equal a 0.0002 [Wb] and RMS error of estimated torque equal a 0.0005 [Nm]), this estimator is implemented on the Virtex-4 FPGA board with minimal resources(less 5% of slices registers, 5% of DSP48Es and around 17% of LUTS). After the validation step our estimator was used to achieve a DTC control in a process of hardware co-simulation. The spectral analysis of the speed curve obtained by "Powergui FFT Analysis Tool" from Matlab prove that the ripple of the torque and of the flux generated by the estimation errors does not influence the dynamic response of the speed.

Finally, we believe that the proposed estimator reduces the difficulties of the estimation phase for the DTC command and their derived commands. In addition, our estimator can be used for other tasks such as diagnosis of the induction motor.

As a forthcoming application, we will take advantage of the results achieved in this work in order to use our estimator for the implementation of SVM-DTC on FPGA.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Abdelghani Aib; methodology, Djalal Eddine Khodja; software, Abdelghani Aib; validation, Loutfi Benyettou, and Salim Chakroune; formal analysis, Salim Chakroune; investigation, Abdelghani Aib; resources, Djalal Eddine Khodja; data curation, Loutfi Benyettou; writing—original draft preparation, Abdelghani Aib; writing—review and editing, Djalal Eddine Khodja; visualization, Salim Chakroune; supervision, Loutfi Benyettou; project administration, Djalal Eddine Khodja.

References

[1] H. Sudheer, S. Kodad, and B. Sarvesh, "Improvements in direct torque control of induction motor for wide range of speed operation using fuzzy logic", *Journal of Electrical Systems and Information Technology*, Vol. 5, No. 3, pp. 813-828, 2018.

[2] I. Takahashi and Y. Ohmori, "High-performance direct torque control of an induction motor", *IEEE Transaction on Industrial Application*, Vol. 25, No. 2**,** pp. 257-264, 1989.

[3] M. Depenbrock, "Direct Self Control of Inverter-Fed Induction Machines", *IEEE Transaction on Power Electronics*, Vol. 3, No. 4, pp. 420-429, 1988.

[4] H. H. Vo, D. Q. Nguyen, Q. T. Nguyen, C. T. Dong, T. C. Tran, and P. Brandstetter, "Pulse-width modulation direct torque control induction motor drive with Kalman filter", *TELKOMNIKA*, Vol. 19, No. 1, pp. 277-284, 2021.

[5] T. Sutikno, N. R. N. Idris, and A. Jidin "A New Fixed Switching Frequency Direct Torque Controlled PMSM Drives with Low Ripple in Flux and Torque", *ITB Journal of Engineering Science*, Vol. 43, No. 3, pp. 173-190, 2011.

[6] B. Bossoufi, M. Karim, A. Lagrioui, and S. Lonitta, "Performance Analysis of Direct Torque Control (DTC) for Synchronous Machine Permanent Magnet (PMSM)", In: *Proc. of IEEE 16th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pp. 237-242, 2010.

[7] N. Farah, M. H. N. Talib, Z. Ibrahim, Q. Abdullah, O. Aydogdu, Z. Rasin, A. Jidin, and J. M. Lazi, "Analysis and investigation of different advanced control strategies for high-performance induction motor drives", *TELKOMNIKA*, Vol. 18, No. 6, pp. 3303-3314, 2020.

[8] C. L. Toh, N. R. N. Idris, A. H. M. Yatim, "Design and Implementation of TMS320C31 DSP and FPGA for Conventional Direct Torque Control (DTC) of Induction Machines", *International Journal of Mechanical and Mechatronics Engineering*, Vol. 03, No. 2, pp. 225-230, 2009.

[9] L. G. Su, L. D. Hyun, Y. T. Woong, L. K. Beum, S. J. Ho, and C. Ick, "Speed and Flux Estimation for an Induction Motor Using a Parameter Estimation Technique", *International Journal of*

Control, Automation, and Systems, Vol. 3, No. 1, pp. 79-86, 2005.

[10] T. Sutikno, N. R. N. Idris, and A. Z. Jidin, "Overview on Strategies and Approaches for FPGA Programming", *TELKOMNIKA*, Vol. 12, No. 2, pp. 273-282, 2014.

[11] T. Porselvi, K. Deepa, and R. Mutha, "FPGA Based Selective Harmonic Elimination Technique for Multilevel Inverter", *International Journal of Power Electronics and Drive System (IJPEDS)*, Vol. 9, No. 1, pp. 166-173, 2018.

[12] M. Selvarathinam, T. Deepika, D. Devipriya, and P. Muthukumar, "A Novel FPGA Based Direct Torque Control for Induction Motor", *Middle-East Journal of Scientific Research*, Vol. 24, No. 3, pp. 787-793, 2016.

[13] C. T. Kowalski, J. Lis, and T. O. Kowalska, "FPGA Implementation of DTC Control Method for the Induction Motor Drive", In: *Proc. of the International Conference on Computer as a Tool EUROCON*, pp. 1916-1921, 2007.

[14] S. Krim, S. Gdaim, A. Mtibaa, and M. F. Mimouni, "FPGA-Based Implementation Direct Torque Control of Induction Motor", *International Journal of Power Electronics and Drive System (IJPEDS)*, Vol. 5, No. 3, pp. 293-304, 2015.

[15] T. Sutikno, N. R. N. Idris, A. Z. Jidin, and M. Z. Daud, "FPGA Based High Precision Torque and Flux Estimator of Direct Torque Control Drives", In: *Proc. of IEEE Applied Power Electronics Colloquium (IAPEC)*, pp. 122-127, 2011.

[16] N. Kabache, S. Moulahoum, and H. Houassine, "Hardware Co-Simulation of an Adaptive Field Oriented Control of Induction Motor", *Journal of International Conference on Electrical Machines and Systems*, Vol. 3, No. 2, pp. 116-120, 2014.

[17] D. E. Khodja, S. Simard, and R. Beguenan, "Implementation of Optimized Approximate Sigmoid Function on FPGA Circuit to use in ANN for Control and Monitoring", *Control Engineering and Applied Informatics Journal*, Vol. 17, No. 2, pp. 64-72, 2015.

[18] T. Sutikno, A. Z. Jidin, A. Jidin, and N. R. N. Idris, "FPGA Based High Performance Torque and Flux Estimator", *International Review of Electrical Engineering (I.R.E.E)*, Vol. 6, No. 1, pp. 207-214, 2011.

[19] V. A. Bondarko and A. T. Zaremba, "Speed and flux estimation for an induction motor without position sensor", In: *Proc. of the American Control Conference*, pp. 3890-3894, 1999.

[20] S. Boukadida, S. Gdaim, and A. Mtiba, "Hardware Implementation of FTC of Induction Machine on FPGA", *Electronics*, Vol. 20, No. 2, pp. 76-84, 2016.

[21] A. Mohammedi, N. Kabache, S. Moulahoum, and H. Houassine, "FPGA Hardware In the Loop Validation of Direct Torque Control for Induction Motor", In: *Proc. of 20th International Conference on Method and Models in Automation and Robotics (MMAR)*, pp. 812-816, 2015.

[22] J. E. Volder, "The CORDIC Trigonometric Computing Technique", *IRE Transaction on Electronic Computers*, Vol. EC-8, No.3, pp. 330-334, 1959.

[23] J. S. Walther, "A Unified Algorithm for Elementary Functions", In: *Proc. of Spring Joint Computer Conference*, pp 379-385, 1971.

[24] T. Sutikno, N. R. N. Idris, A. Jidin, and M. N. Cirstea, "An Improved FPGA Implementation of Direct Torque Control for Induction Machines", *IEEE Transactions on Industrial Informatics*, Vol. 9, No. 3, pp. 1280-1290, 2013.

[25] M. A. Zare, R. G. Kavasseri, and C. Ababei, "FPGA-based Design and Implementation of Direct Torque Control for Induction Machines", In: *Proc. of Internationl Conference on ReConfigurable Computing and FPGAs (ReConFig14)*, pp. 1-6, 2014.

[26] O. S. Hernandez, J. R. Magdaleno, R. M. Caporal, and E. B. Huerta, "HIL simulation of the DTC for a three-level inverter fed a PMSM with neutral-point balancing control based on FPGA", *Electrical Engineering*, pp. 1441-1454, 2018.

[27] E. Lotfi, M. Elharoussi, and E. Abdelmounim, "VHDL design and FPGA implementation of direct torque control for induction machines", *Bulletin of Electrical Engineering and Informatics*, Vol. 10, No. 3, pp. 1220-1231, 2021.