FPGA Based High Performance Torque and Flux Estimator

Tole Sutikno¹, Aiman Zakwan Jidin², Auzani Jidin³, Nik Rumzi Nik Idris⁴

Abstract – This paper presents a new design of the torque and stator flux estimators for Direct Torque control (DTC) for Field Programmable Gate Array (FPGA) implementation, which permit very fast calculations. An alternative variable word-size approach in two’s complement fixed-point format is used for the implementation, in order to minimize calculation errors and the hardware resource usage. The simulation results of DTC model in Matlab, which performed double-precision calculations, are used as references to digital computations executed in FPGA implementation. The Hardware-in-the-loop (HIL) method is used to verify the minimal error between Matlab simulation and the experimental results, and thus the well-functionality of the implemented estimators. Copyright © 2011 Praise Worthy Prize S.r.l. - All rights reserved.

Keywords: Direct Torque Control, FPGA, HIL, MATLAB/Simulink, Torque and Flux Estimator

Nomenclature

ADC Analog to digital converter
DSP Digital signal processing
DTC Direct torque control
FOC Field oriented control
FPGA Field programmable gate array
HIL Hardware-in-the-loop
Ia Ib a and b components of the stator current in abc reference frame
Ia* Ib* D and Q components of the stator current in stationary reference frame
P Number of pole pairs
φs Stator flux
φD φQ D and Q components of the stator flux in stationary reference frame
R Rotor resistance
S0 S1 S2 Switching states of phase a, b, c
Tm Electromagnetic torque
Tc Sampling time
VD VQ D and Q components of the stator voltage in stationary reference frame
VDC DC link voltage
VHDL VHDL hardware description language
VHSIC Very-high-speed integrated circuits

I. Introduction

DTC scheme has gained popularity to be implemented for motor drives. The main reason for its popularity is due to its simple structure, especially when compared with FOC scheme, which was introduced a decade earlier. This scheme is well known as robust control as it is less dependency on machine parameters, without the complex field orientation block, speed encoder and the inner regulation loop [1]-[13].

Since DTC was first introduced by Takahashi (1986) [14] and Depenbrock (1988) [15], several variations to its original structure were proposed to overcome the inherent drawbacks of DTC such as, inconstant switching frequency, torque ripple, and high sampling requirement for digital implementation [2]-[8], [16]-[18].

For digital implementation, the DTC algorithm is frequently implemented in a Microcontroller or DSP [2], [4], [12], [17], [19]-[23]. However, serial calculations are performed and therefore, they cannot execute fast computation without any losses. As an adequate solution, fast calculations are performed by using FPGA [24]-[29]. Moreover, the high sampling time is the crucial part which allows the minimization of torque ripple [17], [24]-[29].

However, it is not easy to implement DTC in FPGA hardware. The hardest part in the DTC implementation is the torque and flux estimations [24]-[25], [27]-[30]. Toh [17] has implemented all parts of the DTC in FPGA hardware, unless torque and flux estimator part. As a result, she only succeeds to make the sampling time limited to 55μs. Actually, Monmasson [24] has developed the implementation of DTC in FPGA hardware, but because it using third party (MATLAB/Simulink with the Xilinx System Generator fixed-point toolbox), so the sampling time limited to 50μs. Ferreira [31] also has difficulty to increase the sampling frequency, and he only succeeds to improve till 25μs. In the other research, despite an increasing of processing time till 20.34μs has been achieved by Utsumi [32], but the sampling time of the DTC still 25μs. References [20]-[22] were worse; they use the sampling frequency 150, 100 and 100μs respectively.

This main contribution of the research presents a new design of the torque and flux estimator of DTC for FPGA implementation, using two’s complement fixed-point representation with variable word sizes through the
estimation process, and to improve the sampling time till 5μs. The validation of the implementation is done by verifying the experimental results during the motor’s steady state.

II. Direct Torque Control

Figure 1 represents the topology of a DTC drive. The estimated flux magnitude and torque are compared with their references values. Torque and flux comparators are consists of three and two-level hysteresis respectively. Besides, the sector judgment evaluates the position of the stator flux vector in DQ coordinates.

![Figure 1. DTC topology](image)

The switching table produces the switching statuses according to the output of torque and flux comparators and the sector judgment. Those switching statuses are connected to the inverter, which is connected to the motor. They are also used as the input for torque and flux estimation.

To estimate the stator flux and the electromagnetic torque, several parameters need to be determined. Firstly, the stator currents from the motor $I_a$ and $I_b$, are transformed into DQ coordinates, which are adequate to DTC algorithm, as follows:

$$I_D = I_a$$  \hspace{1cm} (1)

$$I_Q = \frac{\sqrt{3}}{3}(I_a + 2I_b)$$  \hspace{1cm} (2)

At the same time, by using the switching status ($S_a$, $S_b$ and $S_c$) produced by the switching table, the stator voltages in DQ components are determined:

$$V_D = \frac{V_{dc}}{3} \left(2S_a - S_b - S_c \right)$$  \hspace{1cm} (3)

$$V_Q = \frac{\sqrt{3}}{3} V_{dc} \left(S_b - S_c \right)$$  \hspace{1cm} (4)

Then, using the calculated $I_b$, $I_a$, $V_D$ and $V_Q$, the estimation of the stator flux in DQ coordinates are performed as follows:

$$\varphi_D = \varphi_{D_{ref}} + (V_D - R_S I_D) T_s$$  \hspace{1cm} (5)

$$\varphi_Q = \varphi_{Q_{ref}} + (V_Q - R_S I_Q) T_s$$  \hspace{1cm} (6)

Notice that $R_s$ is the estimated stator resistance, while $T_s$ is the implementation sampling time. In addition, equations (5) and (6) correspond to the integration using Back Euler Method. As a matter of fact, [21]-[22] suggested that a filter should be added to the integrator in the practical implementation. Thus, equation (5) and (6) become:

$$\varphi_D = \left( \varphi_{D_{ref}} + (V_D - R_S I_D) T_s \right) (1 - \omega_s T_s)$$  \hspace{1cm} (7)

$$\varphi_Q = \left( \varphi_{Q_{ref}} + (V_Q - R_S I_Q) T_s \right) (1 - \omega_s T_s)$$  \hspace{1cm} (8)

Finally, equation (9) calculates flux magnitude by using a square root calculation, whereas the electromagnetic torque is estimated in equation (10):

$$\varphi_e = \sqrt{\varphi_D^2 + \varphi_Q^2}$$  \hspace{1cm} (9)

$$T_e = \frac{3}{4} P (I_Q \varphi_D - I_D \varphi_Q)$$  \hspace{1cm} (10)

III. FPGA Based Proposed Torque and Flux Estimator Architecture

The algorithm of torque and flux estimation is implemented in an architecture consisted of five main blocks, as shown in Figure 2. This architecture has six inputs: two 21-bit currents $I_a$ and $I_b$ and three switching status $S_a$, $S_b$ and $S_c$. At the end, it produces four outputs: the estimation values of torque ($T_e$), flux ($\varphi_e$) and sector. The sampling time choise is 5 μs, which is limited by the ADC used.

![Figure 2. Block Diagram of torque and flux estimators](image)

All the equations which are modeling the motor behavior is implemented in a two-stage-pipelined architecture, as presented in Figure 3. Several mathematical operations are performed in parallel. At the first stage, stator currents and voltages in DQ-coordinates are calculated in parallel so that those results can be used to estimate the stator flux in the same stage.

The resulted currents and flux are used to determine the flux magnitude and the torque estimation in the second stage. A 62-bit non-restoring square root is implemented in order to compute the flux magnitude.
As the matter of fact, [17] proposed that three-stage-pipelined architecture should be implemented in this module, by separating the computation of stator currents and voltages from the estimation of the stator flux. However, the former can be considered as an immediate calculation and thus, those calculations can be merged into one single stage. As a consequence, the latency of the estimator is reduced from 15 µs to 10 µs.

To achieve a good implementation, several digital characteristics need to be considered when designing this estimator. Adopted binary format, quantization and sampling time are among those key factors.

III.1. Binary Format

In this implementation, two’s complement fixed-point representation is used during all the operations, except for the square root calculation. In that particular case, unsigned fixed-point representation is applied, since its operand and its results are always positive.

Recent DTC implementation generally used 32-bit format where some bits might be left unused, while the 16-bit format is not appropriate to achieve good DTC implementation [23]. Therefore, variable word-size approach is adopted for this implementation and so, all the redundant bits can be eliminated by truncating process to minimize the hardware resources usage.

III.2. Quantization

The determination of word size is one of the critical parts in FPGA implementation. On one hand, insufficient number of bits used may reduce the precision or cause the calculation error, which can unnormalize the whole system. On the other hand, larger words used may increase the hardware area used for the implementation.

Since two’s complement fixed-point format is used for the implementation, at least 2 things need to be verified. Firstly, the size of the integer must be properly chosen to avoid the problem of overflow. Secondly, the number of the fractional bits used must be sufficient in order to minimize the quantization errors.

For example, due to the fact that the input currents \( I_s \) and \( I_b \) are varied from -10A to 10A, at least 5 bits are necessary for the integer bits. While 16 fractional bits used can result in a very good precision, since the resolution is very small (=15 µA).

One of the critical parts in this architecture is the stator flux estimation, where the integration is performed. This operation can easily produce errors if the sampling time \( T_s \) is not properly scaled. In this case, \( T_s = 5 µs = 0.000005 \) s. In fact, a minimum of 21 bits is necessary to represent \( T_s \). In this case, \( T_s = 0.00000476837s \) \( (0.00000000000000000000001010)_2 \). However, 27-bit representation is chosen to have a better precision and thus, \( T_s = 0.00000499934s \) \( (0.000000000000000000000010101111111)_2 \).

Figures 4 show the estimated torque taken during the steady state. From this figure, it shows that the torque estimation for 21-bit Ts is imprecise, when compared to Matlab double precision estimation, which is the ideal case.

In fact, the number of bits is increasing after each operation in order to avoid calculation errors or imprecision. This will result in the rising of the hardware area used. Therefore, truncation process must be performed avoid the excessive increase of the number of bits used.

In the example of \( I_s \) calculation as shown in Figure 5, when \( I_s \) is multiplied by 2, the result should be in 6.16
bits (6 integer bits plus 16 fractional bits). Nevertheless, it is stored in 7.16 bits to avoid overflow, which may happen during the addition operation. Next, when the addition result is multiplied by $\frac{1}{3 \sqrt{3}}$, which coded in 1.18 bits, $I_p$ should be represented in 8.34 bits. But, it is truncated to 6.16 bits.

Fig. 5. The example of $I_q$ calculation

### III.3. Sampling Time

The sampling time $T_s$ is 5 µs. Therefore, all the operations involved in this model were performed within this period.

Notice that the use of high sampling frequency is important in DTC implementation, for the purpose to minimize the torque ripple. The sampling time used for DSP implementation is normally much bigger than $T_s$, which is not less than 50 µs. Therefore, it is reduced by a factor of 10 for this FPGA implementation and thus, lower torque ripple is produced, as shown in Figures 6.

Figs. 6. The effect of sampling time to torque ripple: (a) Estimated torque for $T_s$=5 µs; (b) Estimated torque for $T_s$=50 µs

### IV. Experimental Results

The validation of designed torque and flux comparators was performed based on Hardware-in-the-Loop (HIL) method. The DTC model in Matlab Simulink was simulated and then, the same data $I_a$, $I_b$, $S_a$, $S_b$, and $S_c$ used for the simulation were copied from Matlab workspace to VHDL codes, as the inputs for the targeted FPGA. The VHDL codes were simulated in Modelsim before being synthesized and implemented in Altera DE2 EP2C35F672C6. The test design flow is presented in Figure 7, and the photograph of the hardware experimental setup is shown on Figure 8.

The experiments were executed for three different motor’s speeds: low speed, middle speed and high speed. All the experimental results were compared to the validated Matlab simulation results.

Since the hardware resources in FPGA are limited to store all the inputs and the outputs, the tests were taking place only during certain periods of motor’s steady state. The implementation results were observed on the oscilloscope.

Figure 9 presents the input of the low-speed test while Figures 10 to 12 shows the comparisons between Matlab simulations and the experimental results.

For the middle-speed test, the inputs were set as presented in Figure 13, whereas the comparison between the experimental results and Matlab simulations are shown in Figures 14 to 16.

Finally, in the high-speed test, the inputs were set as presented in Figure 17 and the comparison between the experimental results and Matlab simulations are shown in Figures 18 to 20.
Figs. 9. The inputs for low-speed test
(a) Stator Currents Ia and Ib; (b) Sa; (c) Sb; (d) Sc

Figs. 10. Comparison of the torque estimation in low-speed test;
(a) Matlab simulation, (b) the experimental result

Figs. 11. Comparison of the flux estimation \( (\varphi_d, \varphi_q) \) in low-speed test;
(a) Matlab simulation, (b) the experimental result

Figs. 12. Comparison of the flux locus in low-speed test;
(a) Matlab simulation, (b) the experimental result

Figs. 13. The inputs for middle-speed test
(a) the stator currents Ia and Ib; (b) Sa; (c) Sb; (d) Sc

Figs. 14. Comparison of the torque estimation in middle-speed test;
(a) Matlab simulation, (b) the experimental result
The experimental results shown were corresponded well with the simulation in Matlab, which was done in double-precision computation. Besides, the implementation of the algorithm was also validated for different motor’s speed. It can be observed by comparing the form of the estimated torque triangles or the number of flux waveform complete cycle visualized within the same period, for example.

V. Conclusion

FPGA is an alternative for the realization of a high-performance DTC implementation owing to its high processing frequency, which is a reliable device to be utilized for any high-speed execution digital signal processing algorithm which cannot be obtained by any DSP application. The choice of word sizes, the binary format and the sampling time used are very important in order to achieve a good implementation of the estimators. This paper has presented high performance torque and
flux estimator of DTC based on FPGA using two’s complement fixed-point representation through the estimation process. The implementation has successfully improved the sampling time till 5µs to reduce the torque and flux ripple for the DTC drives. The experimental results have verified during the motor’s steady state. The design, which coded in synthesizable VHDL, utilized 2093 logic elements for the implementation on Altera EP2C35F672C6 device, producing very precise estimations with minimal quantization and calculation errors.

References

Authors’ information
1Universitas Ahmad Dahlan (UAD), Yogyakarta, Indonesia.
2Ecole Supérieure d’Ingénieur en Electronique et Electrotechnique Paris (ESIEE Engineering), Paris, France.
3Universiti Teknologi Malaysia Melaka (UTM), Melaka, Malaysia.
4Universitas Teknologi Malaysia (UTM), Johor Bahru, Malaysia.

Copyright © 2011 Praveen Worthy Prize S.r.l. - All rights reserved International Review of Electrical Engineering, Vol. 6, N. 1
Tole Sutikno received his B.Eng. and M.Eng. degree in Electrical Engineering from Diponegoro University, Indonesia and Gadjah Mada University, Indonesia, in 1999 and 2004, respectively. Since 2001 he has been a lecturer in Electrical Engineering Department, Universitas Ahmad Dahlan (UAD), Indonesia. Currently, he is pursuing PhD degree at the Universiti Teknologi Malaysia (UTM), Malaysia. His research interests include the field of power electronics, motor drive systems, and FPGA applications.

E-mails: tsutikno@ieee.org
tole@ee.uad.ac.id

Aiman Zakwan Jidin is M.Sc candidate of Ecole Supérieur d’Ingénieur en Electronique et Electrotechnique Paris (ESIEE Paris). His research interests include the Field Programmable Gate Array (FPGA) application and digital electronic design. On July 2007, he is accepted for one month internship at Phoenix Mecano France, July-August 2008 and May-August 2010 are accepted for assistant Engineer Internship at Universiti Teknologi Malaysia (UTM) Skudai, Johor, Malaysia and Jan-June 2011 is accepted for an Engineer Internship at Logic Design Solutions, France.

E-mail: jidin@esiee.fr

Auzani Jidin received his B.Eng. degree and M.Eng. degree in Power Electronics & Drives from Universiti Teknologi Malaysia (UTM), Malaysia in 2002 and 2004, respectively and currently PhD Student in the same University. He is a lecturer in Department of Power Electronics and Drives, Faculty of Electrical Engineering at Universiti Teknikal Malaysia Melaka (UTeM), Malaysia. His research interests include the field of power electronics, motor drive systems, FPGA and DSP applications.

E-mails: auzani@ieee.org
auzani@ute.edu.my

Nik Runzi Nik Idris received the B.Eng. degree in Electrical Engineering from the University of Wollongong, Australia, the M.Sc. degree in power electronics from Bradford University, West Yorkshire, U.K., and the Ph.D. degree from Universiti Teknologi Malaysia (UTM) in 1989, 1993, and 2000, respectively. He is an Associate Professor at the Universiti Teknologi Malaysia, and an Administrative Committee Member of the Industry Applications Societies/Power Electronics/Industrial Electronics Joint Chapter of IEEE Malaysia Section. His research interests include ac drive systems, FPGA and DSP applications in power electronic systems.

E-mail: nikrunzi@ieee.org