Speed Control for Brushless DC Motors using PID Algorithm

Abstract

Field Programmable Gate Arrays (FPGA) has gained attraction in the field of Industrial automation and is becoming more popular in motor control applications due to its robustness and customizability.
FPGA (Field Programmable Gate Arrays)는 산업 자동화 분야에서 큰 호응을 얻었으며 견고함 및 사용자 정의 가능성으로 인해 모터 제어 응용 분야에서 널리 보급되고 있습니다.

The use of FPGA provides outstanding hardware design flexibility.
FPGA를 사용하면 탁월한 하드웨어 설계 유연성을 제공합니다.

This further enables implementation of a fully customizable peripheral set and also helps to meet the increasing performance requirements.
이로써 완벽하게 사용자 정의 할 수 있는 주변 장치 세트를 구현할 수 있으며 증가하는 성능 요구 사항을 충족하는 데 도움이 됩니다.

The paper covers the use of Xilinx Kintex 7 FPGA based implementation of close loop control system.
이 논문은 close loop control system의 Xilinx Kintex 7 FPGA 기반 구현의 사용을 다룹니다.

The close loop system controls the speed of Three-Phase Brushless Direct Current (BLDC) motor, by using feedback from three hall sensors.
close loop system은 3 개의 hall sensor로부터 feedback을 사용하여 3상 Brushless Direct Current (BLDC) motor의 속도를 제어합니다.

Xilinx Kintex 7 FPGA based speed controller of brushless DC motor system is designed and implemented using PID algorithm.
Xilinx Kintex 7 FPGA 기반의 brushless DC motor system은 PID algorithm을 사용하여 설계되고 구현된다.

The design is implemented using Verilog Hardware Description Language (HDL).
설계는 Verilog Hardware Description Language(HDL)을 사용하여 구현됩니다.

As FPGA supports reconfigurable computing and provides a facility of on-chip parallelism, good operational performance was achieved.
FPGA가 재구성 가능한 컴퓨팅을 지원하고 on-chip 병렬 처리 기능을 제공함에 따라 우수한 운영 성능이 달성되었습니다.

Keywords

BLDC, PID, PWM, RPM, FPGA

Introduction

Nowadays, brushless DC motors are widely used in various applications, right from motor vehicle, and industrial application to aircrafts for the control application.
요즘, brushless DC motor는 자동차, 산업용 애플리케이션에서 제어 애플리케이션을 위한 항공기에 이르기까지 다양한 애플리케이션에 널리 사용되고 있습니다.

The primary reason of increasing usage of brushless DC motor can be attributed to good weight/size to power ratio, excellent acceleration performance, little or no maintenance and less acoustic and electrical noise than Brushed DC motors.
brushless DC motor의 사용 증가의 주된 이유는 Brushed DC motor보다 중량 / 크기 대 전력 비율이 우수하고 가속 성능이 우수하며 유지 보수가 거의 또는 전혀 없고 음향 및 electrical noise가 적기 때문입니다.

This has led to significant researches for the speed control on the BLDC motors.
이로 인해 BLDC 모터의 속도 제어에 대한 중요한 연구가 이루어졌습니다.

In BLDC motor, the commutation of a BLDC motor is controlled electronically.
BLDC motor에서는 BLDC motor의 정류가 전자식으로 제어됩니다.

To rotate the BLDC motor the stator windings should be energized in a sequence.
BLDC motor를 회전 시키려면 stator winding(권선)이 순서대로 활성화되어야 합니다.

It is important to know the rotor position in order to understand which winding will be energized following the energizing sequence.
어떤 winding(권선)이 energizing sequence에 따라 energized 될지를 이해하기 위해서는 rotor 위치를 알아야 합니다.

Rotor position is sensed using Hall-effect sensors embedded into the stator.
Rotor 위치는 stator(고정자)에 내장 된 Hall-effect sensor를 사용하여 감지됩니다.

Mostly BLDC motors have three Hall sensors inside the stator on the non-driving end of the motor.
대부분의 BLDC motor는 motor의 비구동 단에 있는 stator 내부에 3 개의 Hall sensor를 가지고 있습니다.

Whenever the rotor magnetic poles pass near the Hall sensors they give a high or low signal indicating the N or S pole is passing near the sensors.
Rotor 자극이 Hall sensor 근처를 통과 할 때마다 N 또는 S 극이 센서 근처를 통과 함을 나타내는 높거나 낮음 신호를 제공합니다.

Based on the combination of these three Hall sensor signals, the exact sequence of commutation can be determined.
이 3 개의 Hall sensor의 조합에 따라 정류의 정확한 순서를 결정할 수 있습니다.

The remainder of the paper is arranged as follows.
논문의 나머지 부분은 다음과 같이 정리되어있다.

Section 2 describes the three phase commutation.
2 절에서는 3 상 정류에 대해 설명합니다.

Section 3 explains different methods for speed control of BDLC motor and provides detailed explanation of PID algorithm.
3 장에서는 BDLC motor의 속도 제어 방법에 대해 설명하고 PID algorithm에 대해 자세히 설명한다.

The implementation of FPGA based speed control of motor using three feedback hall sensors is described in section 4.
3 개의 feedback hall sensor를 사용하는 FPGA 기반 속도 제어의 구현은 4 장에서 설명합니다.

While Section 5 shows the simulation and validation results of the implemented design.
5 장에서는 구현 된 설계의 시뮬레이션 및 유효성 검사 결과를 보여줍니다.

Finally, conclusions are drawn in Section 6 and references are listed in Section 7.
마지막으로 6 절에서 결론을 도출하고 7 절에서 참조를 나열한다.

Three Phase Motor Commutation (3상 모터 정류)

As compared to Brushed motors, the commutator and brushes in Brushless motors are replaced by six Transistor/MOSFETs/IGBTs to form three-phase Bridge as shown in figure1.
Brushless motor와 비교하여 Brushless motor의 commutator(정류기)와 브러시는 그림 1과 같이 6 개의 트랜지스터 / MOSFET / IGBT로 대체되어 3상 브리지를 형성합니다.

Three phase windings either in Star/Delta fashion is connected in motor with each winding 120 degrees apart.
Star/Delta 방식의 3 상 winding(권선)은 120도 간격으로 각 winding(권선)과 함께 모터에 연결됩니다.

Windings are energized by providing a constant voltage level at the transistors on one branch of bridge (as indicated by signals PWM A, PWM B& PWM C);
winding(권선)은 bridge의 한 branch에있는 transistor에서 일정한 전압 레벨을 제공함으로써 활성

화됩니다 (신호 PWM A, PWM B 및 PWM C로 표시됨).

while on the other branch the PWM pulses are applied (as indicated by signals PWM A, PWM B & PWM C).
다른 branch에서는 PWM 펄스가 인가된다 (신호 PWM A, PWM B 및 PWM C로 표시됨).

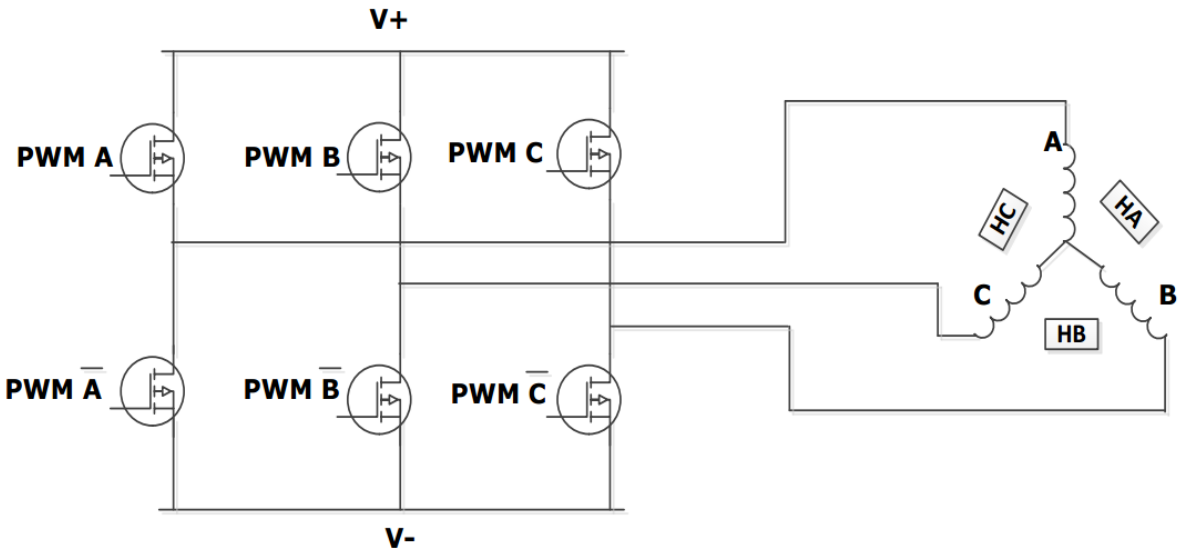

Figure 1 : Bridge Circuit of Motor Driver

그림 1 : motor driver의 브릿지 회로

At any time, one winding is connected to positive terminal through upper transistor;
하나의 winding(권선) 은 언제든지 상단 transistor를 통해 양극 단자에 연결됩니다.

one winding is connected to negative terminal through lower transistor and third winding connected to ground terminal.
하나의 winding(권선)은 접지 단자에 연결된 하부 transistor 및 3개 winding(권선)을 통해 음의 단자에 연결된다.

The phase sequence at each terminal will be as per figure 2.
각 터미널의 phase (위상) sequence(순서)는 그림 2와 같습니다.

Only two phases are active at a time. Hall sensors are used to detect the position of rotor;
한 번에 두 개의 위상 만 활성화됩니다. Hall sensor는 rotor의 위치를 감지하는 데 사용됩니다.

their outputs will appear at exactly 120 degree apart from each other as per figure 2.
이들의 출력은 그림 2와 같이 서로 120도 간격으로 정확하게 나타납니다.

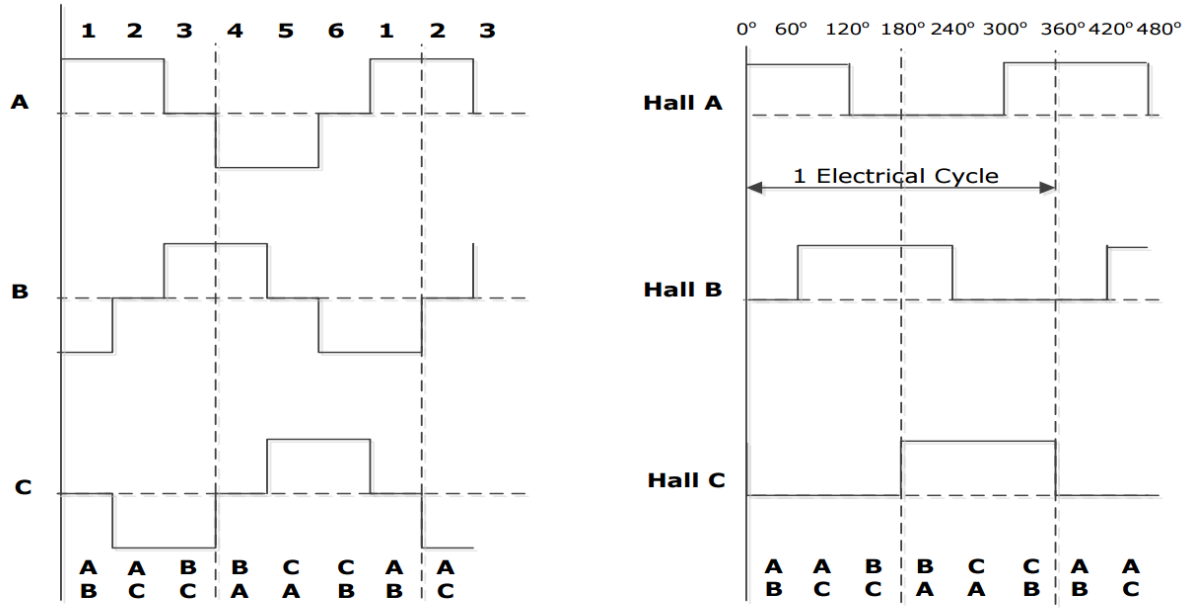

Figure 2 : Three Phase Commutation and Hall Sensor Outputs

그림 2 : 3상 정류 Hall Sensor 출력

Methods for Speed Control of Motor(모터의 속도 제어 방법)

For speed control of BLDC motor using sensor based method, it is required for the controller to know the rotor position for electronic commutation.
센서 기반 방법을 사용하는 BLDC 모터의 속도 제어를 위해서는 컨트롤러가 전자 정류를 위한 rotor 위치를 알아야 합니다.

Sensor based commutation method can be implemented in two ways:
센서 기반 정류 방법은 두 가지 방법으로 구현 될 수 있습니다.

1. Open loop

2. Closed loop

1. Open-loop Method

In this method, the actual motor speed is not tracked and hence there is no feed-back mechanism.
이 방법에서는 실제 모터 속도가 추적되지 않으므로 feed-back mechanism이 없습니다.

Only reference speed is used to control the motor speed by updating PWM duty cycle.
PWM duty cycle을 업데이트하여 모터 속도를 제어하는 데 기준 속도만 사용됩니다.

The advantages of this method are simplified control algorithm and lower cost of implementation.
이 방법의 장점은 제어 알고리즘을 단순화하고 구현 비용을 낮추는 것입니다.

The disadvantage is that accuracy cannot be maintained as there is no way to track the actual motor speed.
단점은 실제 모터 속도를 추적 할 방법이 없으므로 정확성을 유지할 수 없다는 것입니다.

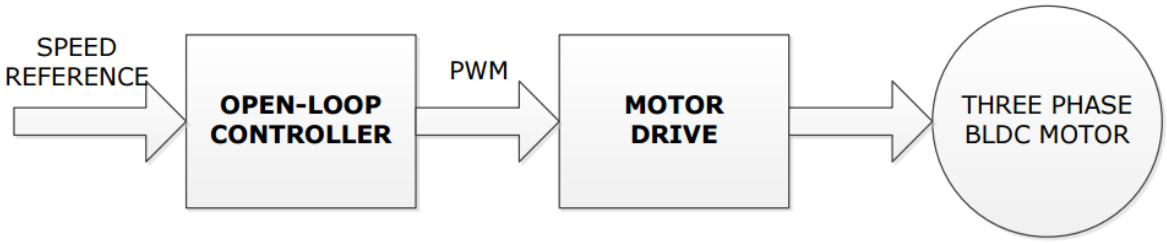

Figure 3: Open Loop Control Method

그림 3 : Open loop 제어 방법

2. Closed-Loop Method

In this method, the actual motor speed is tracked and hence the feed-back mechanism is critical.
이 방법에서는 실제 모터 속도가 추적되므로 feed-back mechanism이 중요합니다.

Here both the reference speed and the actual speed are used to control the motor speed by updating the PWM duty cycle.
여기서 기준 속도와 실제 속도는 PWM duty cycle을 업데이트하여 모터 속도를 제어하는 데 사용됩니다.

The main advantages of this method are the system stability; minimal disturbances compared to open-loop control and reduced sensitivity for dynamic load variations.
이 방법의 주요 장점은 시스템 안정성, open-loop control에 비해 최소한의 교란 및 dynamic load variation(동적 부하 변동)에 대한 감도 감소입니다.

P controller, PI controller, PD controller and PID controller are the controller types which can be used in feedback or feedforward systems based on the system requirements.
P controller, PI controller, PD controller 그리고 PID controller 는 시스템 요구 사항에 따라 feedback 또는 feedforward system에 사용할 수 있는 controller 유형입니다.

PID controller is widely implemented in closed-loop type of feedback control system for speed control of BLDC motors.
PID controller 는 BLDC 모터의 속도 제어를 위한 closed-loop type의 feedback control system에 광범위하게 구현됩니다.

P and PD controller is directly proportional to incoming error;
P 그리고 PD controller 는 들어오는 오류에 직접 비례합니다.

hence a little change in error can cause system instability.
따라서 약간의 오차가 있으면 시스템이 불안정해질 수 있습니다.

PI controller is an accurate and provides good system stability i.e. less steady state error response.
PI controller는 정확하고 양호한 시스템 안정성, 즉 안정된 상태 오차 응답을 제공한다.

But the integral factor in controller takes more iteration to reduce error to zero.
그러나 controller의 적분 요소는 오류를 0으로 줄이기 위해 더 많은 반복을 필요로 합니다.

PID controller is the most reliable, accurate and provides better system stability i.e. less steady state error response as well as it helps to eliminate incoming error to zero with minimum iterations.
PID controller는 가장 신뢰할 수 있고 정확하며 더 나은 시스템 안정성을 제공합니다. 즉, 안정 상태 오류 응답이 적을 뿐 아니라 최소 반복으로 들어오는 오류를 0으로 제거하는데 도움이 됩니다.

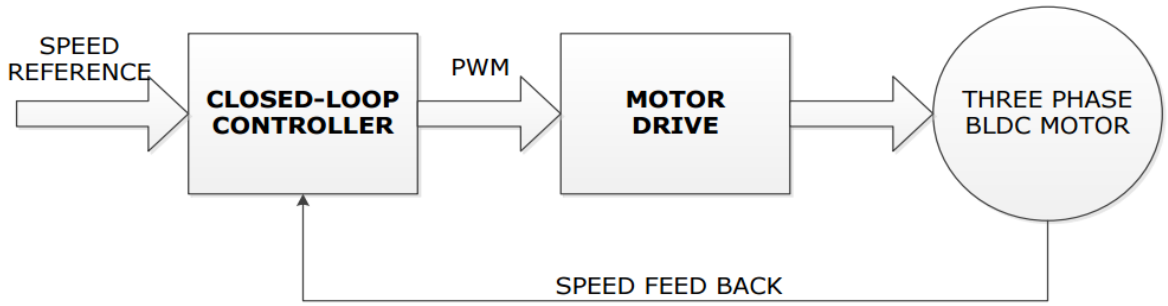

Figure 4 : Closed Loop Control Method

그림 4 : Closed Loop 제어 방법

A. PID Algorithm

PID controller is the most widely used in closed-loop control system.
PID controller는 closed-loop control system에서 가장 널리 사용됩니다.

The algorithm works on the error generated from the difference between the reference speed and the actual speed.
algorithm은 기준 속도와 실제 속도의 차이로 생성된 오류에 대해 작동합니다.

PID parameters Proportional gain Kp, Integral Gain Ki, and Derivative Gain Kd affects system's overall performance.

PID parameters 비례 이득(Proportional gain) Kp, 적분 이득(Integral Gain) Ki 및 미분 이득 (Derivative Gain) Kd는 시스템의 전반적인 성능에 영향을 줍니다.

Hence choosing right parameters for a system is a difficult process and can be done by using several tuning methods which includes manual tuning, Ziegler-Nicholas tuning and Cohen-coon tuning.
따라서 시스템에 적합한 parameters 를 선택하는 것은 어려운 과정이며 manual tuning, Ziegler-Nicholas tuning 및 Cohen-coon tuning 을 포함한 여러 tuning 방법을 사용하여 수행 할 수 있습니다.

Normally a mathematical model of the system is designed along with PID controller and the system performance is observed with applied set of values of PID parameters to finalize the best suited values.
일반적으로 시스템의 수학적 모델은 PID controller와 함께 설계되며 시스템 성능은 적용된 PID parameter 값으로 관찰하여 최적의 값을 결정합니다.

Following are the effects of PID parameters (Kp, Ki, and Kd)
다음은 PID parameter (Kp, Ki 및 Kd)의 영향입니다.

- System Rise time will be reduced by Kp, it provides faster response in variable load condition
- 시스템 상승 시간이 Kp만큼 단축되며 variable load condition (가변 부하 조건)에서보다 빠른 응답을 제공합니다.

- Steady state error will be reduced by Ki , hence the motor speed is pushed near to reference speed
- Ki만큼 Steady 상태 오류가 줄어듦에 따라서 모터 속도가 기준 속도 근처로 밀려납니다

- Settling time and overshoot will be reduced by Kd, hence provides faster response
- 정착 시간과 overshoot은 Kd만큼 줄어들어서 빠른 응답을 제공합니다.

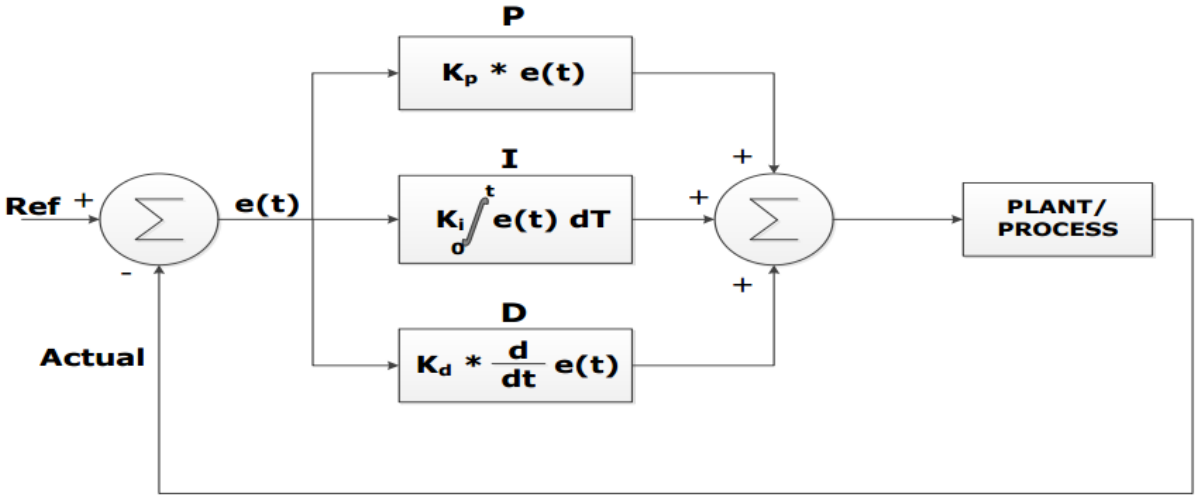

Figure 5 : Block Diagram of a System with PID Controller and Feedback loop

그림 5 : 그림 5 : PID Controller 및 Feedback loop가 있는 시스템의 Block Diagram

PID controller works on following equation:
PID controller는 다음 방정식에서 작동합니다.

$$\mathrm{u(t)} = K_p * e(t) + K_i \int_0^t e(t)dT + K_d * \frac{d}{dt} e(t) \qquad \text{.....(1)}$$

For digital implementation of above continuous form equation, discrete time transformation is required.
위의 연속 형태 방정식의 디지털 구현을 위해서는 이산 시간 변환이 필요합니다.

The discrete form is achieved by first applying Laplace transform on each individual term and subsequently applying z transform on each term.
개별 형태는 각각의 개별 항에 Laplace transform을 적용하고이어서 각 항에 대해 z 변환을 적용함으로써 성취됩니다.

The transformation for each individual term is as shown in table 1.
각 용어에 대한 변환은 표 1과 같습니다.

Table 1: PID Transformations

Term	Time Domain Equation	Laplace Transformation	Z Transformation (Discrete time transformation)
Proportional (P)	$u(t) = K_p * e(t)$	$u(s) = K_p * e(s)$	$u(z) = K_p * e(z)$
Integral (I)	$u(t) = K_i \int_0^t e(t)\, dT$	$u(s) = \frac{K_i}{s} * e(s)$	$u(z) = \frac{K_i}{(1-z^{-1})} * e(z)$
Derivative (D)	$u(t) = K_d * \frac{d}{dt} e(t)$	$u(s) = K_d * s * e(s)$	$u(z) = K_d * (1-z^{-1}) * e(z)$

By replacing values of z transformations from table 1 to equation (1), we get
표 1의 z 변환 값을 방정식 (1)으로 대체하면

$$u(z) = [K_p + \frac{K_i}{(1-z^{-1})} + K_d * (1 - z^{-1})] * e(z) \qquad .....(2)$$

Further simplifying it,(더 단순화,)

$$u(z) = \left[\frac{(K_p + K_i + K_d) + (-K_p - 2K_d) * z^{-1} + K_d * z^{-2}}{(1-z^{-1})}\right] * e(z) \qquad .....(3)$$

By rearranging equation (3) we have(방정식 (3)을 재정렬함으로써)

$$\mathbf{u(z) - z^{-1}\, u(z) = \left[(K_p + K_i + K_d) + (-K_p - 2K_d) * z^{-1} + K_d * z^{-2}\right] * e(z)} \qquad \mathbf{.....(4)}$$

By converting equation (4) in to difference equation, we get following equation, which can be easily implemented in digital systems by using basic digital blocks like adders and multipliers.
방정식 (4)를 차분 방정식으로 변환하면 다음과 같은 방정식을 얻을 수 있습니다. 이 방정식은 가산기 및 곱셈기와 같은 기본 digital block을 사용하여 digital system에서 쉽게 구현할 수 있습니다.

$$\mathbf{u(k) = u(k-1) + (K_p + K_i + K_d) * e(k) - (K_p + 2K_d) * e(k-1) + K_d * e(k-2)} \qquad \mathbf{...(5)}$$

Implementation

Closed loop control method using PID algorithm is implemented on Xilinx Kintex 7 Field Programmable Gate Array (FPGA).
PID algorithm을 이용한 Closed loop control 방식은 Xilinx Kintex 7 Field Programmable Gate Array(FPGA)에서 구현된다.

The top level block diagram for implementation is as shown in figure 6.
구현을 위한 최상위 block diagram은 그림 6과 같습니다.

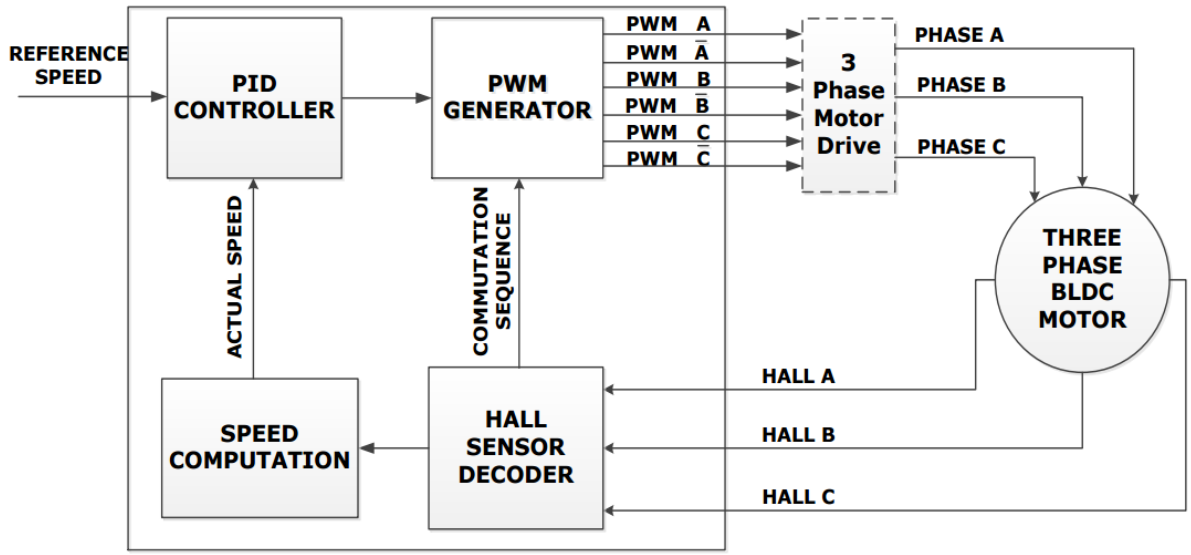

Figure 6 : Implementation Block Diagram

그림 6 : 그림 6 : 구현 블록 다이어그램

1. Hall Sensor Decoder

Different combination of hall inputs will be available based on rotor position.

rotor 위치에 따라 다양한 hall 입력 조합을 사용할 수 있습니다.

Hall input's combinations will repeat in successive manner for an electrical cycle as shown in figure 2.

hall 입력의 조합은 그림 2에서와 같이 electrical cycle 동안 연속적으로 반복됩니다.

By decoding the sequence of hall sensors, we will get information about change in electrical cycle as well as the commutation sequence.

hall sensor의 sequence를 decoding하여 electrical cycle의 변경 및 commutation sequence에 대한 정보를 얻습니다.

2. Speed Computation(속도 계산)

Actual speed of motor is calculated based on electrical cycles.
모터의 실제 속도는 electrical cycle을 기준으로 계산됩니다.

An electrical cycle has direct relation with RPM (Revolution Per Minute) of motor, as an example for single pole motor, one electrical cycle represent one revolution (1 RPM) of motor.
electrical cycle은 모터의 RPM (Revolution Per Minute)과 직접적인 관계가 있으며, single pole

motor 의 예로서 하나의 electrical cycle은 모터의 1회전 (1 RPM)을 나타냅니다.

3. PID Controller

As described in section 3.2.1, PID algorithm is simplified to Equation (5) and is digitally implemented with use of multipliers and adders.
3.2.1 절에서 설명했듯이 PID algorithm은 식(5)로 단순화되며 승수와 덧셈기를 사용하여 디지털로 구현됩니다.

PID processes the incoming error per electrical cycle.
PID는 electrical cycle 으로 들어오는 오류를 처리합니다.

For implementation of this motor control application, the tuning of the PID parameter values (Kp, Ki and Kd) are derived from a mathematical model.
이 motor control application의 구현을 위해, PID parameter 값 (Kp, Ki 및 Kd)의 tuning은 수학적 모델로부터 유도됩니다.

4. PWM Generator

The speed of motor is controlled by varying the motor torque.
모터의 속도는 모터 토크를 변화시킴으로써 제어됩니다.

Motor Torque can be varied by changing average voltage across the motor windings.
motor winding(권선)의 평균 전압을 변경하여 모터 토크를 변경할 수 있습니다.

For varying average voltage across the windings the PWM method is used where the duty cycle of wave is varied according to require speed.
Winding(권선) 양단의 평균 전압을 변화 시키려면 PWM 방식이 요구되는 속도에 따라 파의 duty cycle 이 변하는 곳에서 사용됩니다.

The variation of applied PWM wave is based on hall sensor decoder output to recognize which phase is to be energized.
적용된 PWM 파형의 변화는 hall sensor decoder 출력을 기반으로 어느 위상에 전류가 통하게 될 것인지 인식됩니다.

Here, on one branch; level signal is applied (PWM A^{-} , PWM B^{-} & PWM C^{-}) and on another branch PWM signal is applied (branch PWM A, PWM B and PWM C) which is shown in figure 7.
여기, 한 지점에; (PWM A, PWM B 및 PWM C) 및 다른 분기 PWM 신호가 적용된다 (분기 PWM A, PWM B 및 PWM C). 그림 7에 나와있다.

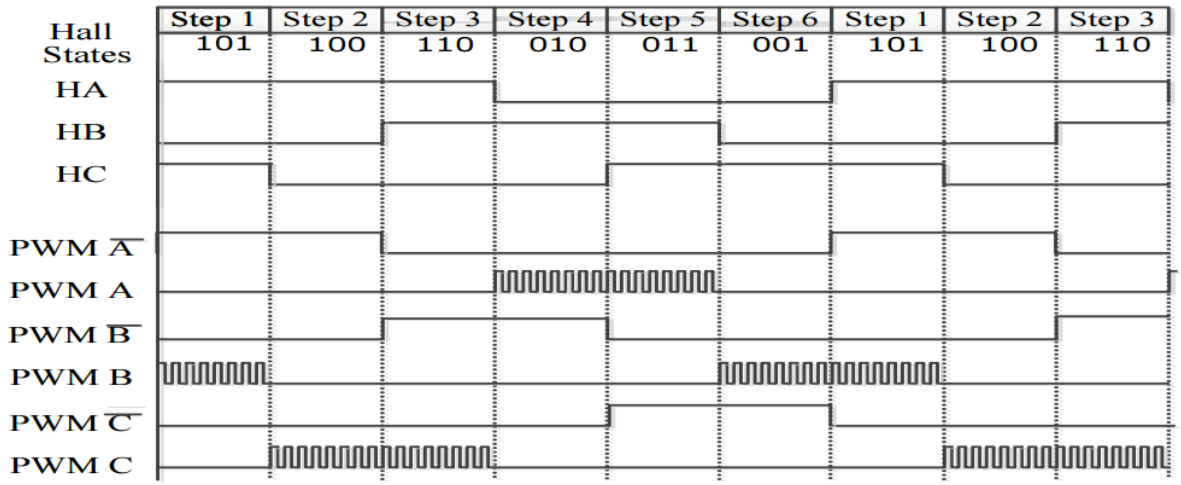

Figure 7 : Commutation sequence based on Hall Sensor

그림 7 : Hall Sensor 기반의 Commutation sequence

5. Implementation Results (구현 결과)

A. Simulation Results

Figure 8 represents simulated waveform for FPGA control logic integrated in system.
그림 8은 시스템에 통합 된 FPGA 제어 로직에 대한 simulated된 파형을 나타냅니다.

Ha, Hb and Hc the hall sensor inputs and reference speed are the test inputs provided through test bench.
Ha, Hb 및 Hc hall sensor 입력 및 기준 속도는 test bench를 통해 제공된 test 입력입니다.

PID continuously processes the input error at every electrical cycle and based on PID output PWM waves are generated for the three phase windings.
PID는 매 test bench마다 입력 오류를 지속적으로 처리하고 PID 출력을 기반으로 PWM 파가 3상 winding(권선)에 대해 생성됩니다.

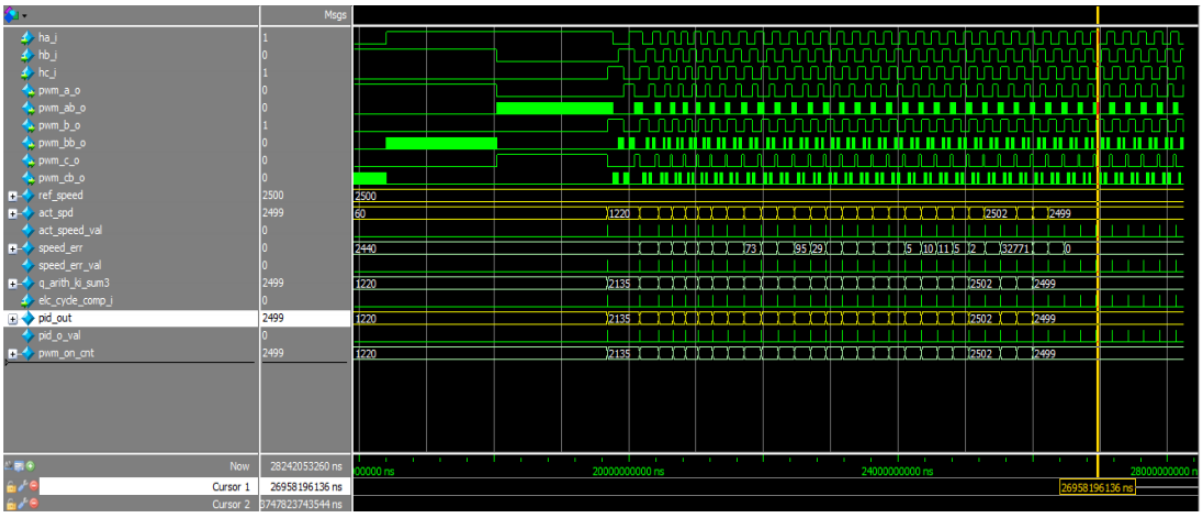

Figure 8 : Simulation Results

그림 8 : Simulation 결과

B. Chipscope Pro Analyzer 결과

Figure 9 represents in chip Xilinx Kintex 7 FPGA debugged waveform using Chip Scope Pro Analyzer for PID control logic.

그림 9는 PID control logic을 위해 Chip Scope Pro Analyzer를 사용하여 칩 Xilinx Kintex 7 FPGA debugged waveform을 나타냅니다.

PID output matches (pid_o) with set reference (ref_speed) after certain electrical cycles.

PID 출력은 특정 electrical cycle 후에 설정된 참조 (ref_speed)와 일치합니다 (pid_o).

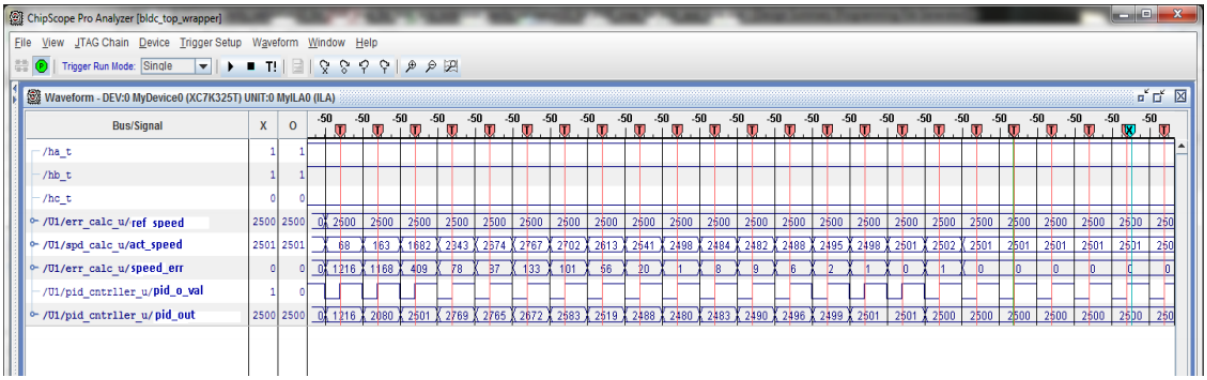

Figure 9 : Chip scope Pro Analyzer Results

그림 9 : Chip scope Pro Analyzer 결과

C. Comparison with mathematical model(수학적 모형과의 비교)

Digital FPGA based implementation with integer arithmetic produces response very close to

mathematical model response as shown in figure 10.
integer arithmetic(정수 산술)을 이용한 디지털 FPGA 기반 구현은 그림 10에서와 같이 수학적 모델 응답에 매우 근접한 응답을 생성합니다.

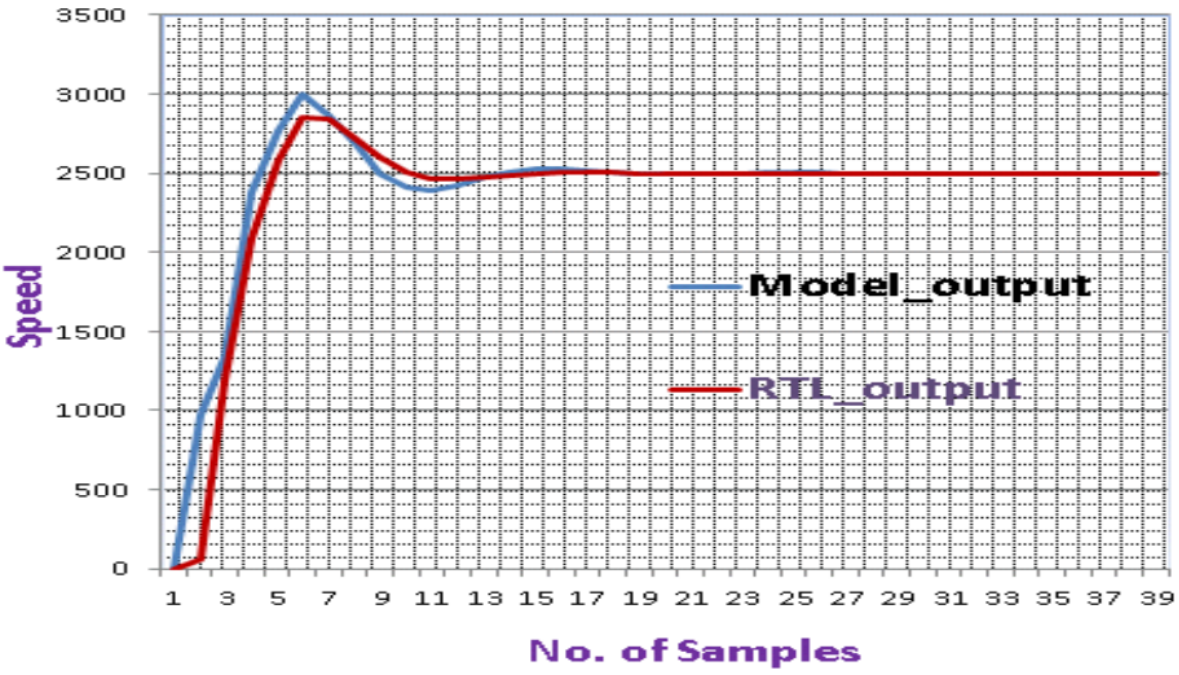

Figure 10 : Model output Vs RTL Implementation output

그림 10 : 모델 출력 Vs RTL 구현 출력

Conclusion(결론)

Through this paper, Hall sensors based closed loop implementation to control the speed of the motor is discussed.
이 논문을 통해 모터의 속도를 제어하는 Hall sensor 기반 closed loop 구현에 대해 논의합니다.

Simulation results and chipscope validation results of implemented PID based BLDC motor controller shows accurate results.
구현 된 PID 기반 BLDC motor controller의 Simulation 결과 및 chipscope 검증 결과는 정확한 결과를 보여줍니다.

Use of PID in the closed loop implementation adds efficiency and stability to systems.
closed loop 구현에서 PID를 사용하면 시스템에 효율성과 안정성이 추가됩니다.

The FPGA based implemented motor controller can be used for system critical applications where

the system stability and motor speed accuracy are the determining factors.
FPGA 기반 구현 motor controller는 시스템 안정성 및 모터 속도 정확도가 결정적인 요소인 시스템에 중요한 application에 사용될 수 있습니다.

References (참고 문헌)

I. slaa503: Texas Instruments Application Report on "Sensored 3-Phase BLDC Motor Control Using MSP430".

II. II. Ishnatek Application Note: "Implementing a Brushless DC Motor Controller on an IGLOO FPGA".

III. III. Dr.Varodom Toochinda, "Digital PID controllers", June 2011