

# 목차

## 1. 설계 목표

## 2. PMSM 제어 이론

- (1) PMSM 개요
- (2) 회전자계
- (3) PID Controller
- (4) dq0 Transform
- (5) CRPWM
- (6) 3phase inveter

## 3. SIMULINK Modeling

- (1) system modeling
- (2) PMSM parameter
- (3) PI controller
- (4) Inverse-dq0 Transform
- (5) CRPWM
- (6) MOSFET Inverter

## 4. Simulation 결과

- (1) PI controller 출력과  $i_{sq}$  출력 비교
- (2) Inverse-dq0 출력 3phase-current reference
- (3) CRPWM 출력 pulse
- (4) MOSFET Inverter 출력 voltage graph
- (5) Stator 3-phase current
- (6) Speed reference, rotor speed, stator  $I_q$

## 5. 고찰/참고

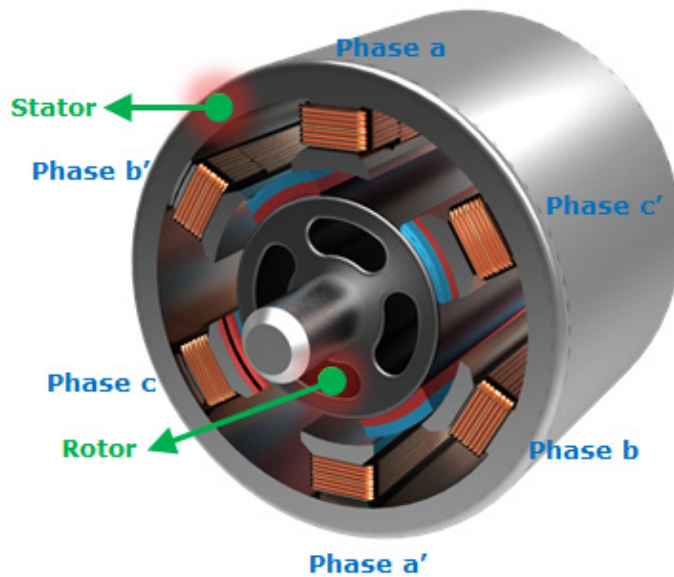
## 1. 설계 목표

PMSM(Permanent Magnet Synchronous Motor)의 원하는 속도로 제어할 수 있는 PMSM controller를 설계하고 정격속도의 절반으로 Reference  $\omega_s$ 를 입력했을 때 결과와 stator  $I_q$ 의 결과를 확인한다.

## 2. PMSM 제어 이론

### (1) PMSM 개요

PMSM(영구자석 동기전동기)는 영구 자석 회전자를 가진 다상 동기기이다.



[그림1] PMSM 단면도<sup>1)</sup>

[그림1]은 3상 영구자석 동기기 단면도이다. 실제 영구자석 동기기는 일정한 계자전류에 의해 여자되며 영구자석 회전자의 실효투자율에 근거한 다양한 인덕턴스를 계산할 수 있다. 소형 영구자석 전동기는, 예를 들면 디스크 드라이브와 팬, 그리고 연료 펌프, 와이퍼, 파워 윈도우, 도어 및 시트 등에 이용되는 자동차산업과 특히 전자산업분야에서 많은 응용을 찾을 수 있다. 대형 영구 자석 기기의 응용 성장은 하이브리드 전기자동차 및 대형 풍력 터빈 발전기보다 더 큰 크기이다. 고정자에 의해 발생하는 공간과 시간 고조파 자속들은 영구 자석 기기의 회전자 구조의 손실을 야기할 수 있지만 자속 손실없이 회전자 자속을 발생하기 때문에 회전자들은 비교적 손실이 없다. 그 결과 에너지 효율의 증시가 증가됨과 동시에 영구자석 기기는 지속적으로 성능을 증가시키는 다양한 응용에서 발견될 것이다.

일반적인 동기기는 전기자 권선은 고정자에, 계자권선은 회전자에 있지만, PMSM은 회전자 구동의 원천으로서 계자권선이 영구자석으로 대체되었다. 전동기는 교류전류가 고정자의 전기자 권선에 공급된다. 전기자 전류에 의해 생성된 자계는 동기속도로 회전하며, 이는 전기자 전류의 주파수에 ( $2/\text{극수}$ )를 곱한 값과 같다.<sup>2)</sup>

1) <https://functionbay.com> PMSM

2) Stephen D. Umans 『전기기기 7<sup>th</sup> Edition』 349p

$$\omega_s = \left(\frac{2}{\text{poles}}\right)\omega_e$$

여기서

$\omega_s$  = 공극 mmf파의 동기 공간 각속도 [rad/sec]

$\omega_e = 2\pi f_e$  = 인가된 전기적 여자의 각 주파수 [rad/sec]

$f_e$  = 인가된 전기적 주파수 [Hz]

일정한 토크는 회전자가 고정자 전류에 의한 자계와 동기 상태로 회전할 때 발생하게 된다. 따라서 동기전동기의 정상상태 속도는 극수와 전기자 전류의 주파수에 의해 결정되며, 일정한 주파수의 교류전원으로부터 여자된 동기 동기전동기는 일정한 속도로 동작하게 된다.

전동기에서 토크는 회전방향으로 발생하며, 기계적 부하를 가동하는데 필요한 반대방향의 부하토크와 균형을 이룬다. 동기전동기의 전기자 전류에 의해 생성된 자속은 회전자의 영구자속 자속보다 앞서 회전하게 되며, 회전자를 당기게 된다.

동기기의 동기속도  $n_s$ [r/min]은 다음과 같이 표현할 수 있다.

$$f_e = \frac{\omega_e}{2\pi} \text{ Hz}$$

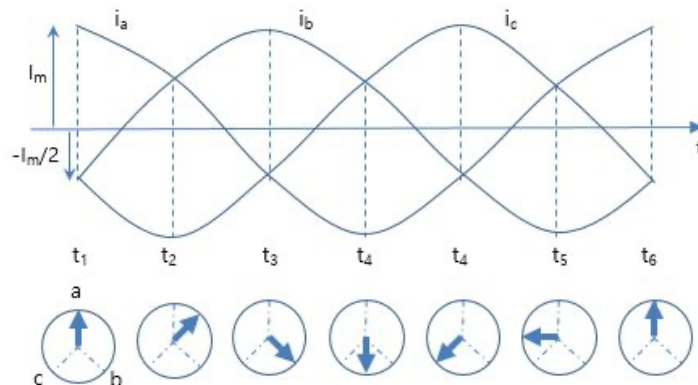
$$n_s = \left(\frac{120}{\text{poles}}\right)f_e \text{ r/min}$$

즉, 동기속도는 극수, 전기자 전류의 주파수에 의해 결정된다는 것을 알 수 있다.

## (2) 회전자계

전기기기의 동작특성은 다양한 권선에 흐르는 전류에 의해 생성되는 자계에 의해 결정된다. 반경  $r_r$ 에서 공극 길이가  $g$ 로 균일한 구조에서 공극 자계  $H$ 는 반경방향으로 향하면서 공극에 따라 균일한 값을 가진다. 따라서 공극 자계  $H_{ag}$ 는 기자력  $F_{ag}$ 에서 공극길이  $g$ 를 나눈 값과 같다.

$$H_{ag} = \frac{F_{ag}}{g}$$



[그림2] 전류에 따른 기자력 파형<sup>3)</sup>

3) [http://www.ktword.co.kr/test/view/view.php?nav=2&id=922&m\\_temp1=5976](http://www.ktword.co.kr/test/view/view.php?nav=2&id=922&m_temp1=5976)

또한  $H_{ag}$ 의 공간적인 기본파 성분은 다음과 같이 표현할 수 있다.

$$H_{ag1} = \frac{F_{ag1}}{g} = \frac{4}{\pi} \left( \frac{N_i}{2g} \right) \cos \theta_a$$

여기서  $N$ 은 코일이 감긴 횟수이다. 그리고 기본파의 크기는 다음과 같다.<sup>4)</sup>

$$(H_{ag1})_{peak} = \frac{4}{\pi} \left( \frac{N_i}{2g} \right)$$

### (3) PI Controller

PI controller(비례적분 제어기)의 전달함수는 다음과 같다.<sup>5)</sup>

$$G_{PI}(s) = K_P + \frac{K_I}{s}$$

그리고 PI의 증가가 응답에 끼치는 영향은 다음과 같다.

PI 이득	오버슈트	정착시간	정상상태오차
$K_P$ 증가	증가	최소 영향	감소
$K_I$ 증가	증가	증가	0(오차를 없앤다)

[표1] PI이득

비례상수와 적분상수를 적절히 조절하면 원하는 출력파형을 얻을 수 있다.

### (4) dq0 Transform

계자권선 축으로 정렬된 직축 요소, 그리고 계자권선 축으로부터 90도 차이나는 횡축 요소, 이 두 요소로부터 동기기글 분석하는 것이 가능하다. 이 개념의 유용성은 회전자의 돌극성 때문에 각각의 고정자의 상에서 시간에 따라 변화하는 인덕턴스로 보이지만, 변환된 양은 회전자와 함께 회전하기 때문에, 일정한 자기경로를 보인다는 점에서 유래한다.

또한 회전자와 고정자의 자속과 기자력 파형의 상호작용을 분석하는 관점에서 유용하다. 고정자 변수를 회전자와 동일하게 회전하는 동일한 변수로 변환함으로써, 정상상태의 이러한 상호작용은 일정한 공간의 각에 의해 분리된 일정한 기자력, 그리고 자속파형과 같은 것이 된다. 이 사실은 회전자 기준에서 관찰하는 것과 일치하는 관점이다.

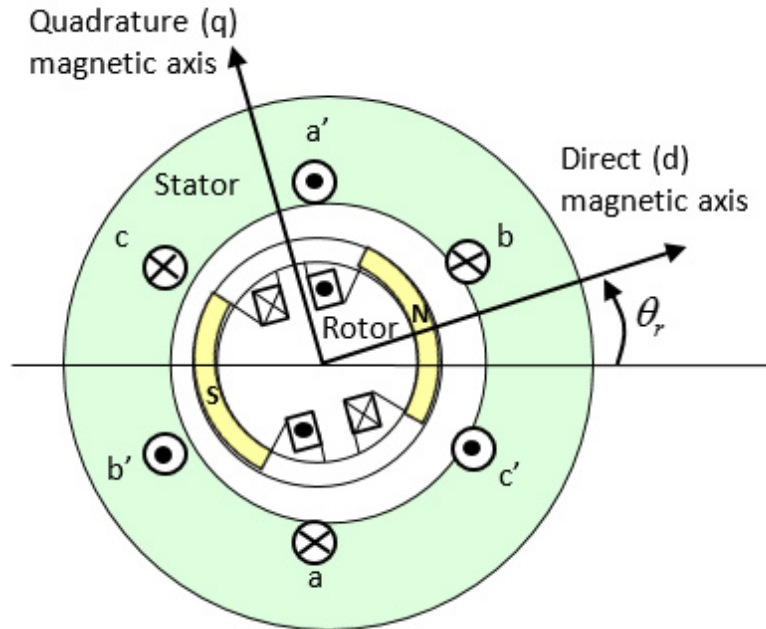
Inverse dq0 transformation은 직축(d-axis), 횡축(q-axis), 중성축(0 axis)을 abc상으로 바꾸는 역할을 한다. 행렬식으로 표현하면 다음과 같다.<sup>6)</sup>

4) Stephen D. Umans, 『전기기기 7<sup>th</sup> Edition』, 235p

5) Richard C. Dorf, 『Modern Control Systems』, 416p

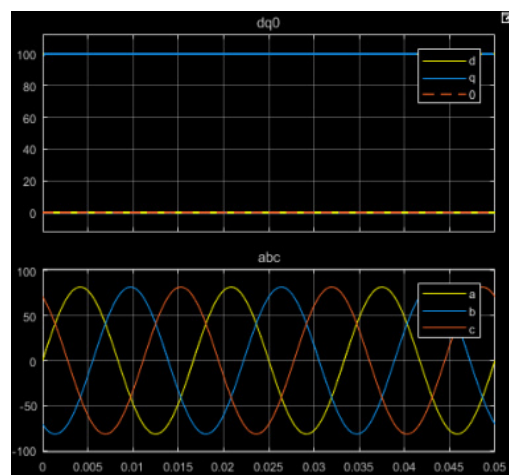
6) Stephen D. Umans, 『전기기기 7<sup>th</sup> Edition』, 739p

$$\begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} = \begin{bmatrix} \cos(\theta_{me}) & -\sin(\theta_{me}) & 1 \\ \cos(\theta_{me} - 120^\circ) & -\sin(\theta_{me} - 120^\circ) & 1 \\ \cos(\theta_{me} + 120^\circ) & -\sin(\theta_{me} + 120^\circ) & 1 \end{bmatrix} \begin{bmatrix} S_d \\ S_q \\ S_0 \end{bmatrix}$$



[그림3] 이상적인 동기기기

이것은 시간에 따라 회전하는 d축, q축, 0축을 표현할 수 있다. 다음 사진은 변환된 abc상으로 변환된 dq0이다.



[그림4] dq0 scope<sup>8)</sup>

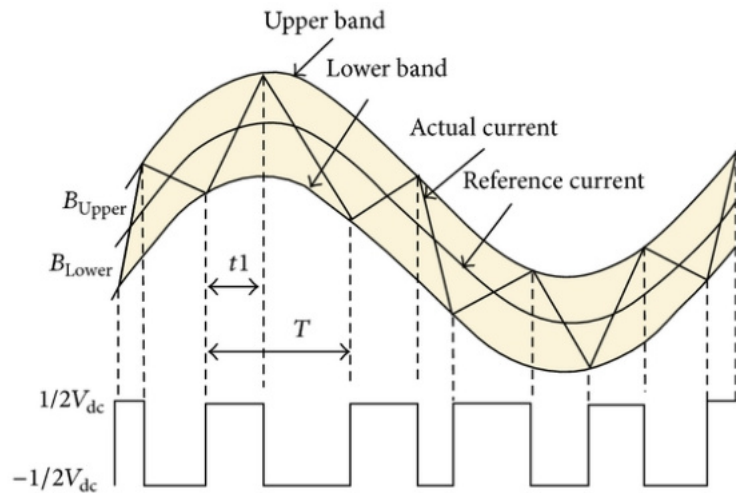
이 프로젝트에서는  $i_d=0$  이다.

7) <http://infoupdate.org/synchronous-motor-stator/>

8) <https://kr.mathworks.com/help/physmod/sps/ref/parktransform.html>

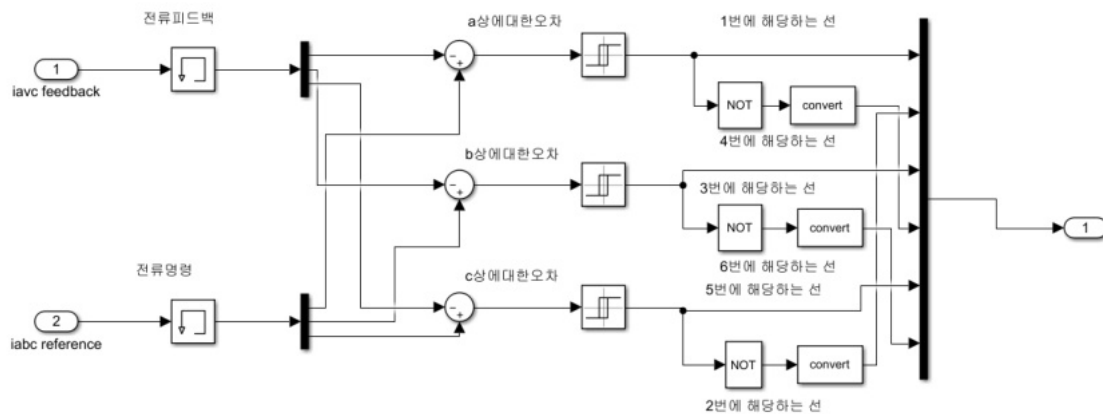
### (5) CRPWM(Current Regulated Pulse Width Modulated)

CRPWM은 전류를 제어하기 위한 소자로 PMSM의 현재 전류와 inv-dq0로부터 나온  $I_{abc}$  reference를 비교하는데 쓰인다. 전류 제어는 Hysteresis control 방식을 쓰는데, 밴드폭을 정해 놓고, 전류가 Upper band보다 높아지면 Lower switch가 켜지고, Lower band 보다 낮아지면 Upper switch가 켜진다. 그리고 이 때 신호가 발생하여 MOSFET Inverter로 보내 준다.



[그림5] Hysteresis band<sup>9)</sup>

그리고 다음은 SIMULINK에서 설계한 CRPWM이다.



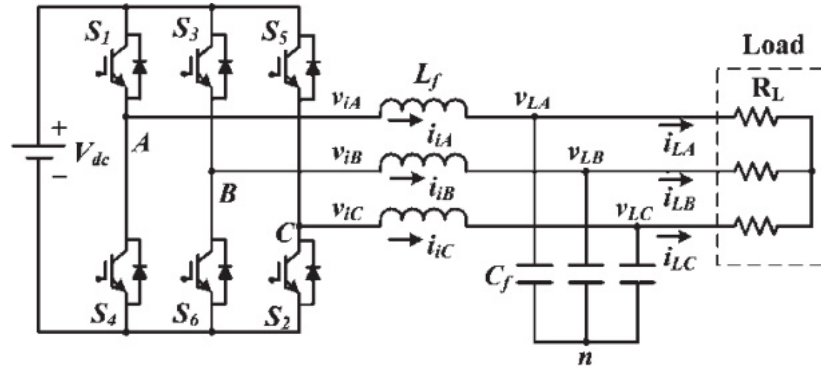
[그림6] CRPWM SIMULNK

[그림6] Relay block에서 Hysteresis band width를 정할 수 있다. 그리고 모터에 있는 inductor로 인해 지연(lagging)되는 현상이 일어나  $I_{ref}$ 와 계산시간이 안 맞는 경우가 생기므로 Memory block을 넣어서 delay를 준다.

9) [https://www.researchgate.net/figure/Diagram-of-hysteresis-current-control\\_fig3\\_277131487](https://www.researchgate.net/figure/Diagram-of-hysteresis-current-control_fig3_277131487)

### (6) 3phase inverter

동기전동기 제어의 간단한 방법 중 하나는 [그림7]에서 보인 3상 인버터와 같은 다상 전압원 인버터에 의하여 전동기를 구동하는 인가 전기자 전압의 주파수를 이용한 속도제어이다.



[그림7] 3phase inverter<sup>10)</sup>

정류기부분은 직류전압  $V_{dc}$ 을 축전기 위에서 만들고, 인버터 부분의 스위치 제어는 가변크기의 펄스폭변조(PWM) 교류전압 파형을 만든다. 예를 들어, 전압  $V_{dc}$ 는 정류기 부분의 위상각 제어를 통하여 그 자체가 변할 수 있다. 인버터 출력파형의 주파수는 당연히 인버터 스위치들의 스위칭 주파수의 제어에 의하여 변할 수 있다. 교류-기기 적용을 위하여, 이 주파수와 연계된 제어는 인가된 전압의 크기제어가 되어야 한다.

10) <http://mwb-online.co/cissell-wiring-diagrams/>



### 3. SIMULINK Modeling

#### (1) system modeling

The image shows the 'Solver Configuration' dialog box in SIMULINK. It is divided into three main sections: 'Simulation time', 'Solver selection', and 'Solver details'.  
1. 'Simulation time': 'Start time' is set to 0.0 and 'Stop time' is set to 1.  
2. 'Solver selection': 'Type' is set to 'Fixed-step' and 'Solver' is set to 'ode4 (Runge-Kutta)'.  
3. 'Solver details': 'Fixed-step size (fundamental sample time)' is set to 1e-5.  
4. 'Tasking and sample time options': 'Periodic sample time constraint' is set to 'Unconstrained'. There are four unchecked checkboxes: 'Treat each discrete rate as a separate task', 'Allow tasks to execute concurrently on target', 'Automatically handle rate transition for data transfer', and 'Higher priority value indicates higher task priority'.

[그림8] system modeling

Solver는 ode4(Runge-Kutta)을 택했다. Runge-Kutta 방법은 적분 방정식중 초기값 문제를 푸는 방법 중 하나이다. Runge-Kutta 방법 따로 설정해 준 이유는 sine wave를 좀 더 부드럽게 해주기 위함이다.

Stop time은 1sec로 설정했다. 설계 목적 중 하나는 0.5초가 settling time인지 확인하는 것인데, SIMULINK를 처음 시작할 때 기본값인 10초로 해버리면 컴퓨터가 계산하는데 오래 걸린다. 1sec도 충분하므로 위와 같이 설정했다.

Fixed-step size는  $1e-5(10^{-5})$ 로 정했다. 얼마나 계산을 자주 해주나 설정하는 것인데, 크기가 작을수록 그래프에 나오는 파형이 부드럽게 나온다. 그러나  $1e-6$ 으로 설정했을 때 노트북 성능이 부족해 파형이 너무 천천히 나온다. 그래서  $1e-6$ 는 포기하고 Fixed-step size를  $1e-5(10^{-5})$ 로 설정했다.

## (2) PMSM parameter

PMSM parameter는 PMSM의 환경설정을 특정한 값으로 정할 수 있다. 본인은 1번 Preset을 하도록 배정받았다.

1번 Preset은 정격토크 0.8Nm, 인가전압은 300Vdc 그리고 정격속도는 3000RPM이다.

Number of phases:  
3

Back EMF waveform:  
Sinusoidal

Rotor type:  
Round

Mechanical input:  
Torque Tm

Preset model:  
01: 0.8 Nm 300 Vdc 3000 RPM - 0.8 Nm

Measurement output  
☐ Use signal names to identify bus labels

[그림9] PMSM Configuration

또한 PMSM parameter도 확인할 수 있다. PMSM parameter는 위 configuration에서 Preset model을 정했으므로 고정된 값이다.

Configuration Parameters

Machine parameters  
Compute from standard manufacturer specifications.

Stator phase resistance Rs (ohm): 18.7

Armature inductance (H): 0.02682

Machine constant  
Specify: Flux linkage established by magnets (V.s)  
Flux linkage: 0.1716

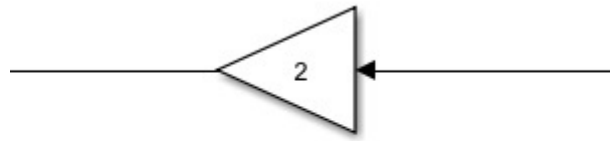
Inertia, viscous damping, pole pairs, static friction [ J(kg.m^2) F(N.m.s) p() Tf(N.m) ]: 1.6e-05 1.349e-05 2 0

Initial conditions [ wm(rad/s) thetam(deg) ia,ib(A) ]: [0,0, 0,0]

Rotor flux position when theta = 0:  
90 degrees behind phase A axis (modified Park)

[그림10] PMSM Parameter

위 항을 보면 pole pairs가 2임을 볼 수 있다. PMSM에서 나온 결과는 기계각속도인데 Inverse dq transform에 feedback할 때 전기각속도로 해야 한다. pole pairs가 2이므로 poles 은 4이다. 그러므로  $(poles)/2 = 2$ 를 Gain을 통해 곱해줘서 feedback 해야 한다.



기계각속도를전기각속도로 바꿔준다(poles/2)

[그림11]  $\omega_s$  to  $\omega_e$

### (3) PI controller

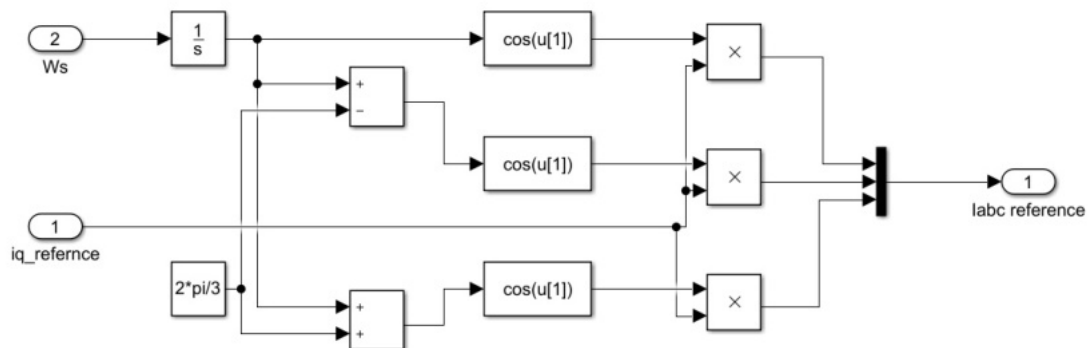
다음 그림은 PI controller parameter이다.

[그림12] PI Controller parameters

P gain( $K_P$ )은 0.00055이고 I gain( $K_I$ )은 0.00303 이다. Overshoot=0과 Settling time=0.5sec 조건을 만족하기 위해서 저러한 값을 택했다. 위 값을 찾는 방법은 하나하나씩 입력하면서 숫자를 줄여나간 것 밖에 없다.

### (4) Inverse-dq0 Transform

다음 그림은 설계한 Inverse-dq0 transform이다.

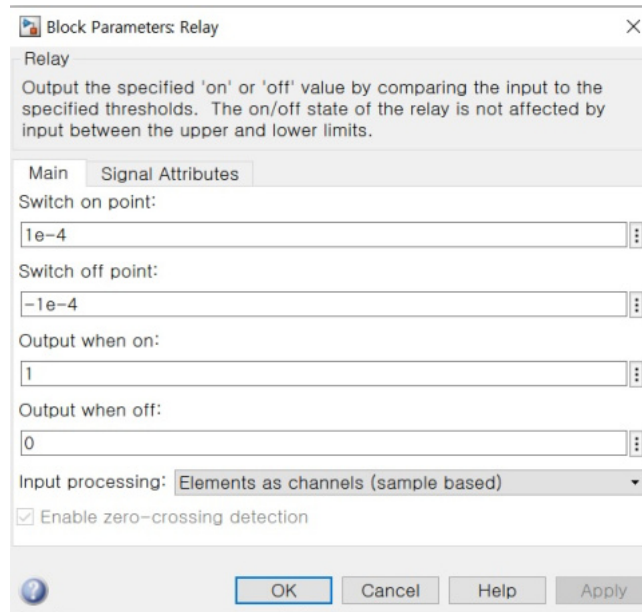


[그림13] Inverse-dq0 Transform

$$\begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} = \begin{bmatrix} \cos(\theta_{me}) & -\sin(\theta_{me}) & 1 \\ \cos(\theta_{me} - 120^\circ) & -\sin(\theta_{me} - 120^\circ) & 1 \\ \cos(\theta_{me} + 120^\circ) & -\sin(\theta_{me} + 120^\circ) & 1 \end{bmatrix} \begin{bmatrix} S_d \\ S_q \\ S_0 \end{bmatrix}$$

a, b, c상에 위상차를 주기위해  $2\pi/3$ 를 b상에는 빼주고 c상에는 더해주었다. 그리고  $\cos(u[1])$  함수는 Fcn block을 이용해 구현했다. 그리고 feedback으로 들어온  $\omega_s$ 에 적분인자를 곱해  $\theta_{me}$ 로 만들었다.

##### (5) CRPWM

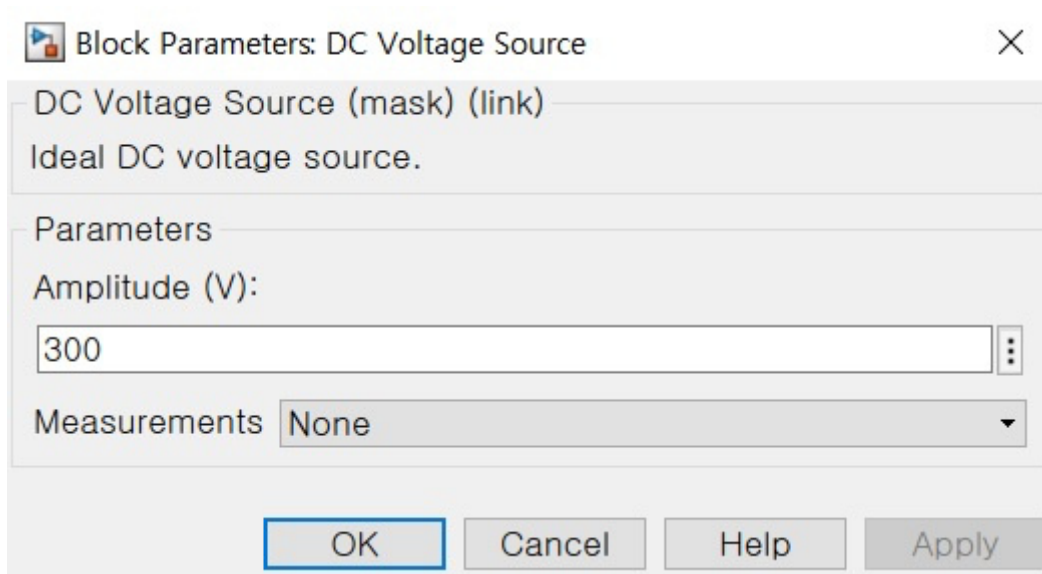


[그림 14] Hysteresis band width

Hysteresis band width을 1e-4, -1e-4로 설정했다. 더 작은 값으로 설정하면 좀 더 직선에 가까운 파형을 관찰할 수 있다.

#### (6) MOSFET Inverter

다음은 MOSFET에서  $V_{dc}$  값을 설정한 모습이다.

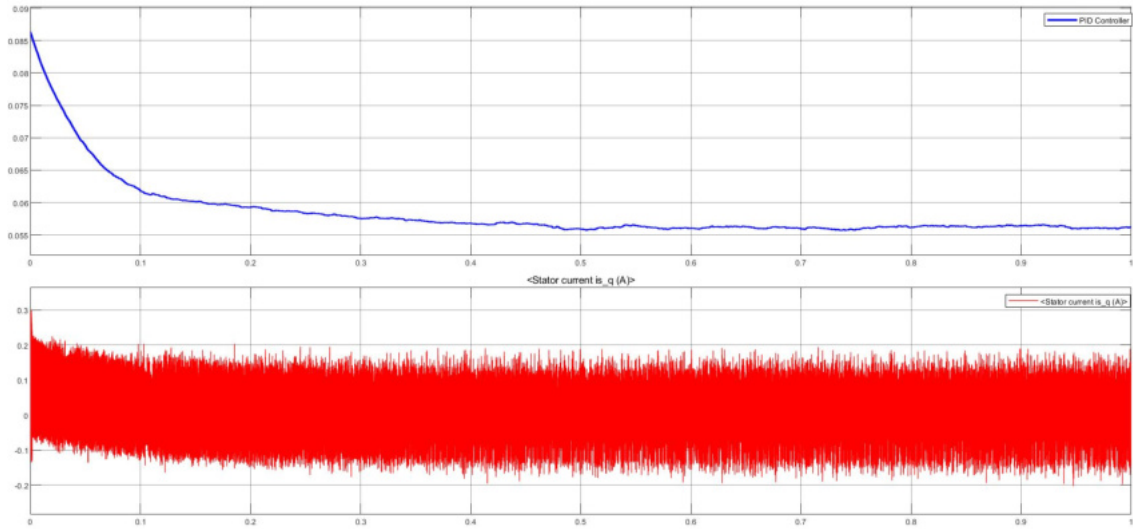


[그림 15]  $V_{dc}$

[그림9]에서 보았듯이 인가전압이 300Vdc이다. 그러므로 Amplitude를 300V로 했다.

## 4. Simulation 결과

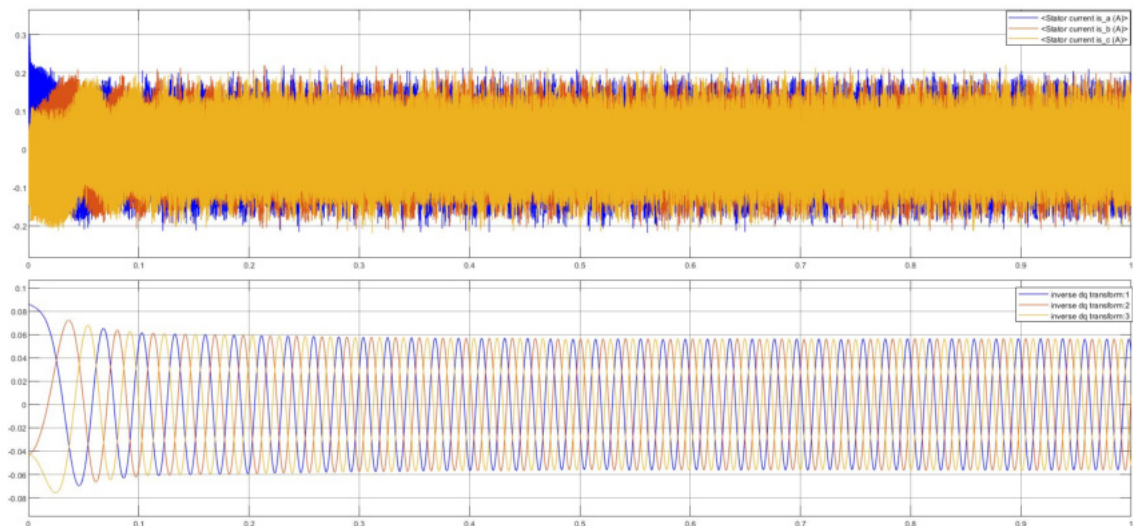
### (1) PI controller 출력과 $i_{sq}$ 출력 비교



[그림16] PI controller,  $i_{sq}$

[그림16]를 보면 PI controller에서 나온 파형과  $i_{sq}$ 에서 나오는 파형이 대략적으로 일치 한다는 점을 볼 수 있다. 즉,  $i_{sq}$ 는 PI controller에게 제어 받고 있다는 사실을 알 수 있다.  $i_{sq}$ 가 저렇게 흔들리는 것은 CRPWM에서 전류 제어를 받고 있기 때문이다. 전체적인 출력인  $i_{sq}$ 을 바꾸고 싶다면 PI controller를 통해 바꾸면 된다.

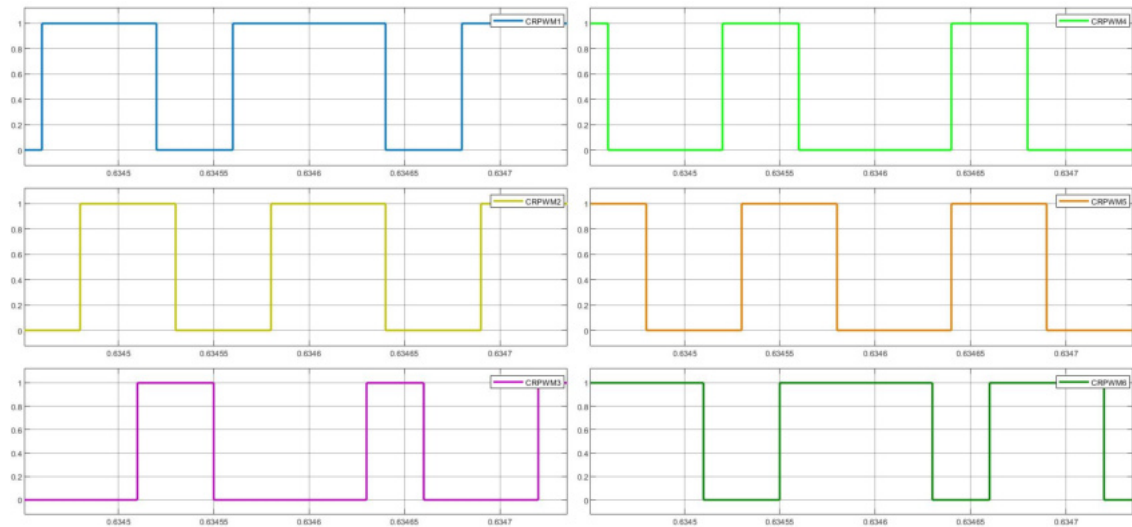
### (2) Inverse-dq0 출력 3phase-current reference



[그림17] Inverse-dq0, 3phase-current reference

[그림17]에서 위 그래프는 Inverse-dq0로 들어가는 feedback 전류이고 아래 그래프는 Inverse-dq0의 출력이다. 두 그래프의 파형을 비교해 보면 굉장히 흡사한 모습을 볼 수 있다. 이는  $I_s$ 가 Inverse-dq0에 의해 제어를 받고 있다는 것을 알 수 있다.

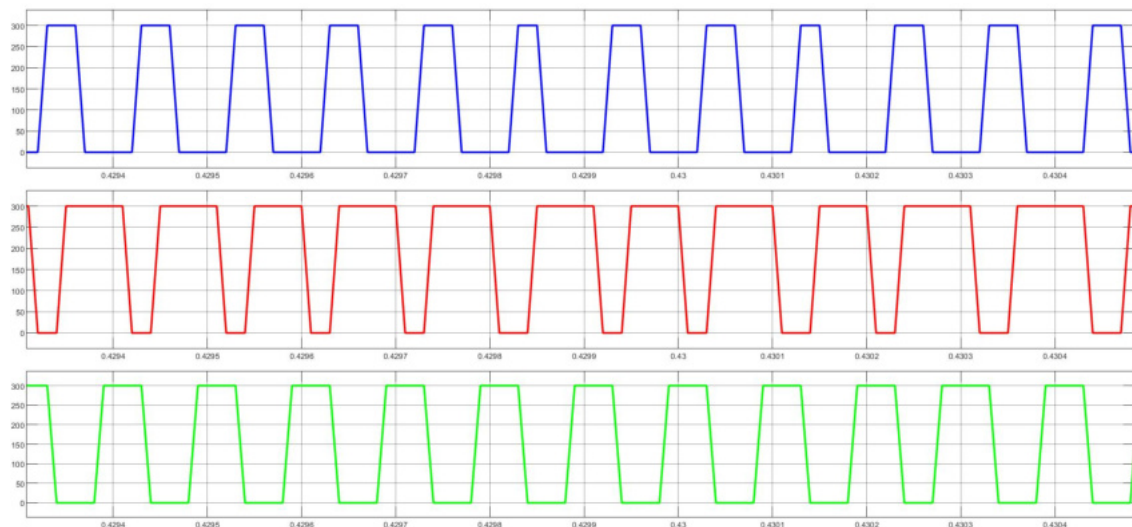
### (3) CRPWM 출력 pulse



[그림18] CRPWM pulse

a상-(1,4), c상-(2,5), b상-(3,6) 으로 묶여있다. 파형을 보면 출력값이 0 또는 1인 것을 알 수 있다. 이는 CRPWM이 각 상마다 스위칭 역할을 해 줌을 볼 수 있다. 묶여있는 그래프끼리 정반대의 모습을 보여주는데 이는 CRPWM의 특징을 보여준다. 전류가 Upper band보다 높을 시 전류를 낮춰주는 스위치를 키고 Lower band보다 작을 시 전류를 높여주는 스위치를 킨다. 그리고 (1,4), (2,5), (3,6) 파형끼리 조금씩 밀려있는 모습을 볼 수 있다. 이는 각 상마다 120° 씩 차이이기 때문이다.

자세히 보기 위하여 MOSFET Inverter에서 전압을 측정했다.

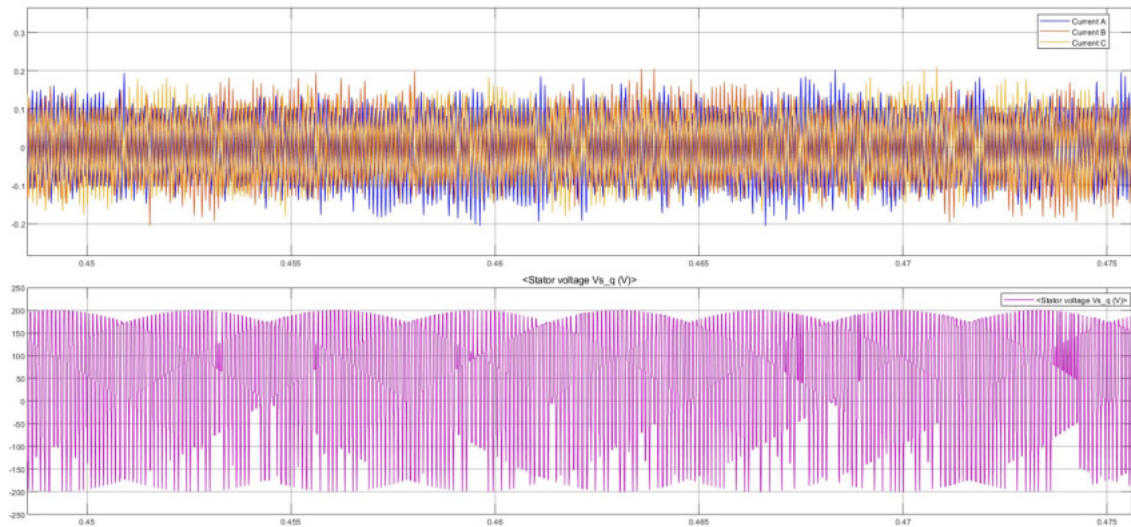


[그림19] MOSFET 내부 전압 스위칭

abc상을 각각 담당하고 있는 MOSFET의 전압을 측정했다. [그림19]을 보면 300V로 상승, 하강을 반복하고 있고 위상차도 나는 것을 볼 수 있다. CRPWM의 역할을 아주 잘 볼 수 있는 모습이다.



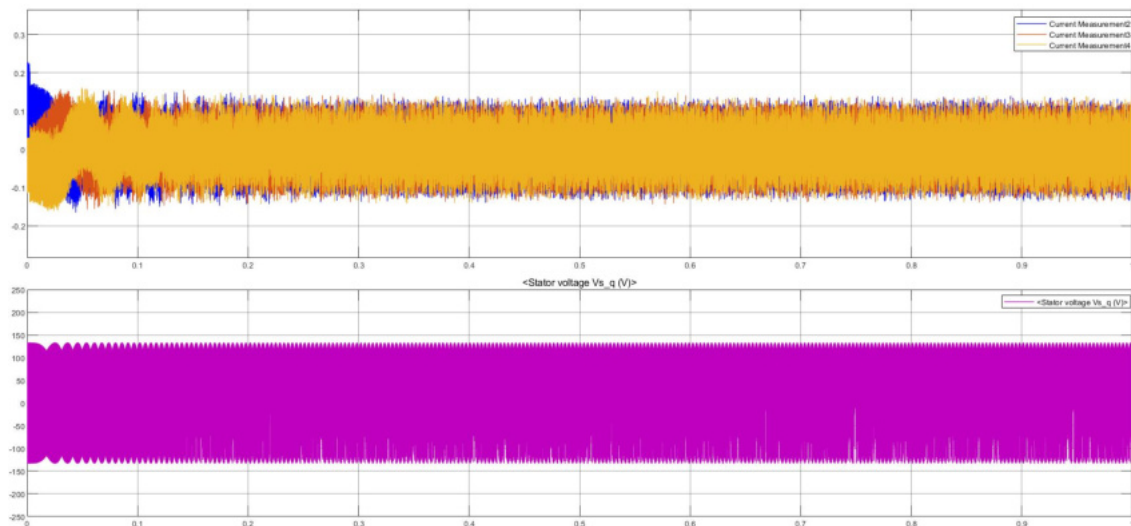
#### (4) MOSFET Inverter 출력 voltage graph



[그림20] MOSFET Inverter 출력 voltage graph

[그림20]에서 위 그래프는 MOSFET Inverter에서 나온  $i_{abc}$ 이고 아래 그래프는 q축에서 측정되는 전압 그래프이다. 아래 그래프를 보면 전압의 Peak값이 포락선을 이루는 것을 볼 수 있다. 이는 3상 전류에 의해 발생된 현상으로 볼 수 있다.

Stator voltage 전압에 영향을 주는 건 무엇인지도 알아보았다. MOSFET에 있는 DC voltage source를 200으로 바꿔 보았다.

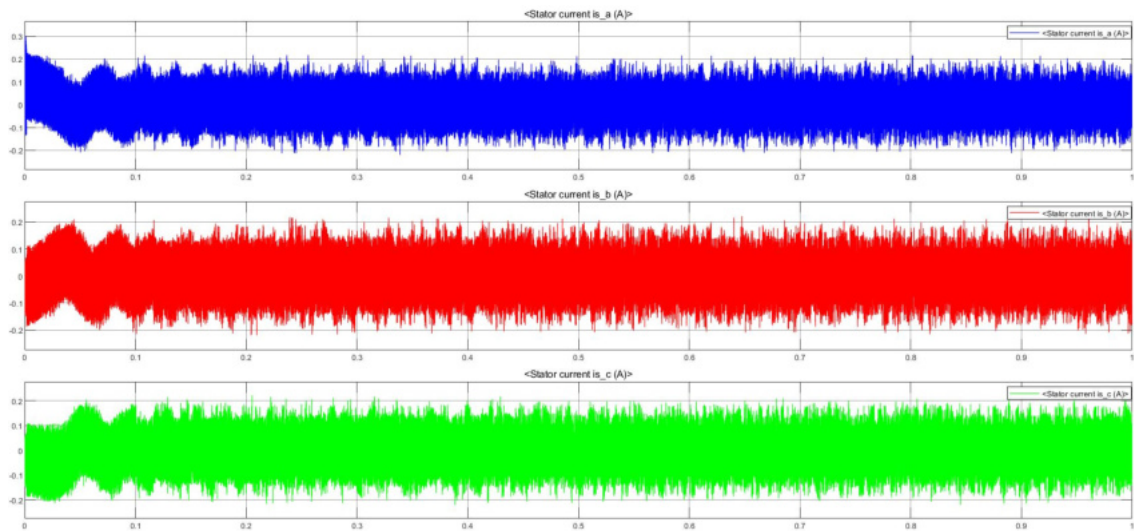


[그림21]  $V_{dc}=200$ 일 때  $V_s$

[그림21]에서 볼 수 있듯이  $V_{dc}$ 를 줄였더니  $V_s$ 값이 줄었음을 볼 수 있다. 그럼 [그림20]에서 측정된 전압은 왜 줄어든 것일까? 이는 PMSM 자체에서 가지고 있는 저항 때문이라고 생각된다. [그림9]를 보면 Stator resistance가 있음을 알 수 있다. 다른 방법으로 PI controller 나 Reference speed를 바꿔 봤지만  $V_s$ 는 변하지 않았다.

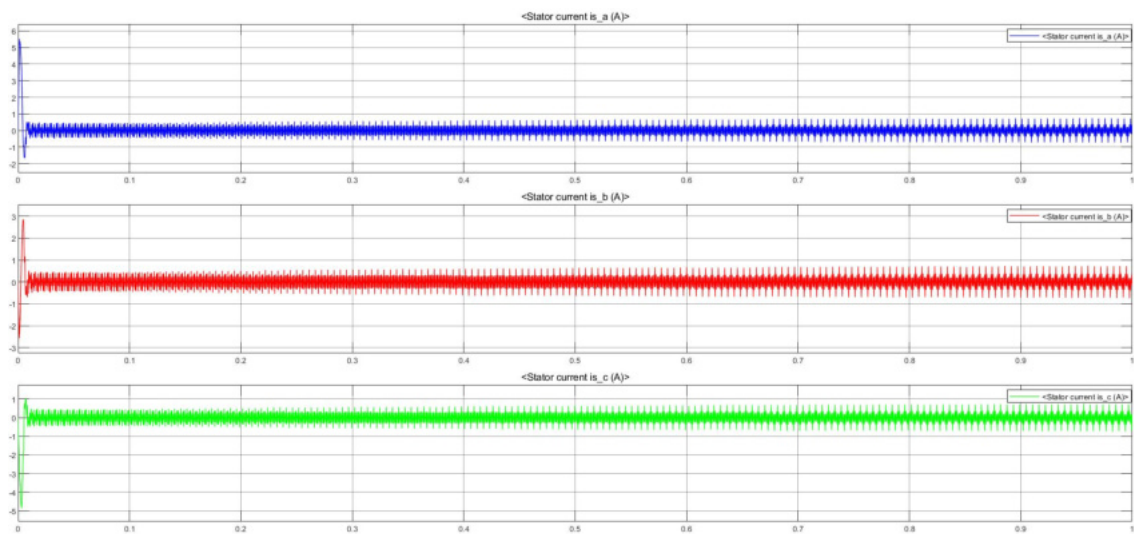


### (5) Stator 3-phase current



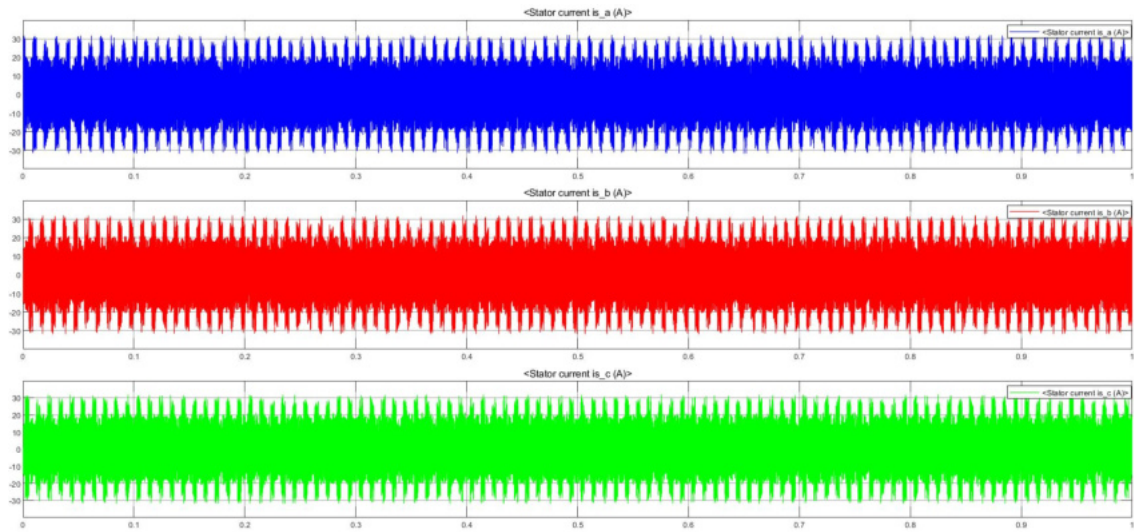
[그림22] Stator 3-phase current

[그림22]는 stator에서 측정된 전류 a, b, c를 분리해서 나타낸 것이다. 전류 abc가  $120^\circ$  씩 위상차가 나므로 똑같은 그래프가 조금씩 밀려있는 모습을 볼 수 있다. 그럼  $I_s$ 의 크기는 어떻게 제어할까? 알아보기 위해 Reference speed값을 극단적으로 10000까지 올려보았다.



[그림23]  $I_s$  when Reference speed=10000

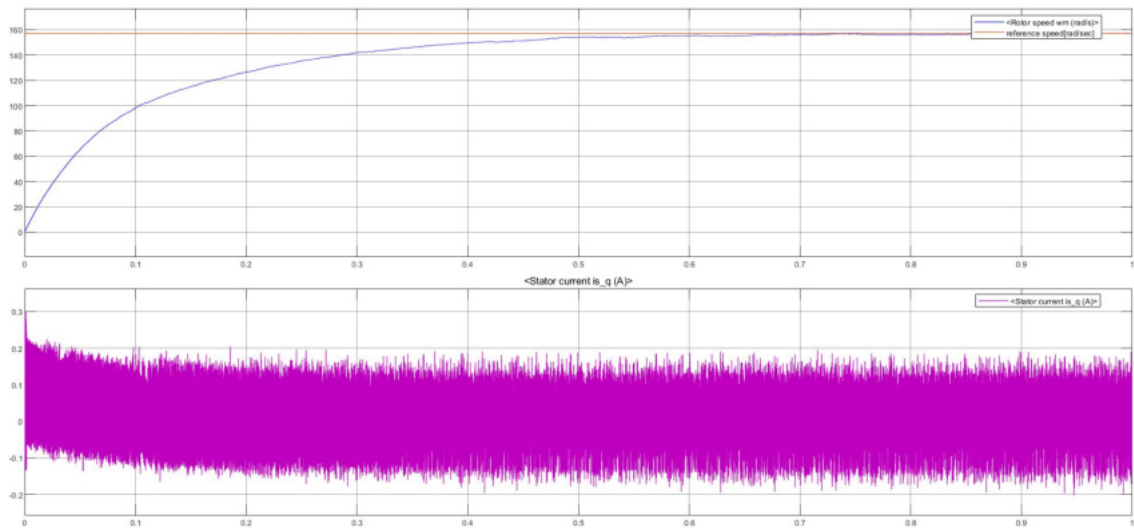
[그림23]를 보면 전류값이 초기에는 크게 늘었으나 금방 정상상태에 돌입하는 것을 볼 수 있다. 이번에는 MOSFET 내부에 있는 DC voltage source를 극단적으로 30000까지 올려 보았다.



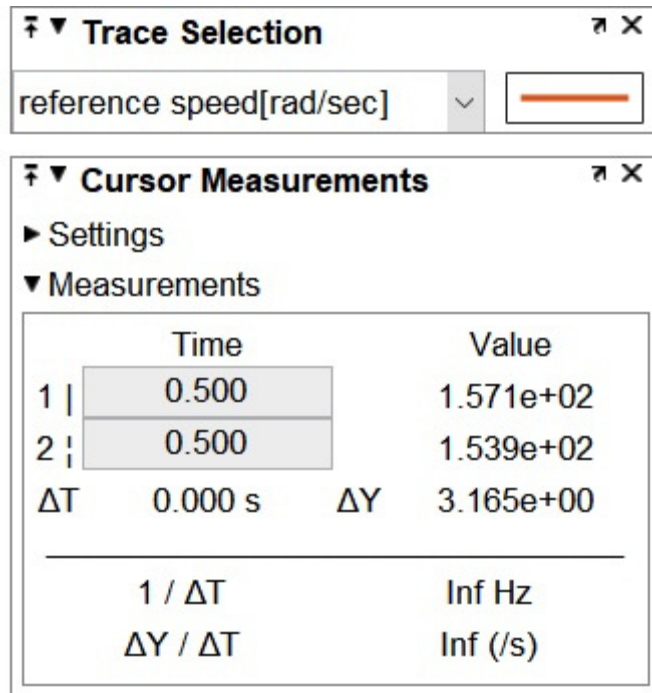
[그림24]  $V_{dc}=30000V$

[그림24]에서는 전류값이 크게 증가한 모습을 볼 수 있다. 이로써  $I_s$ 의 정상상태 값을 정하는 것은  $V_{dc}$ 임을 알 수 있었다. 물론 저렇게 큰 전압이나 전류가 흐르면 모터에 큰 무리가 오므로 하지 않는 것이 좋다.

#### (6) Speed reference, rotor speed, stator $I_q$

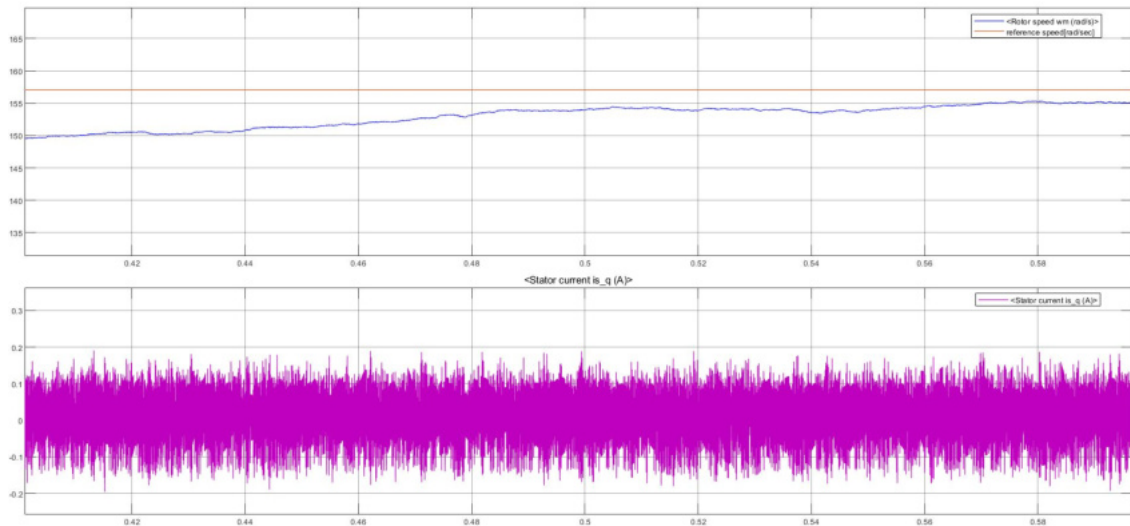


[그림25] 최종결과



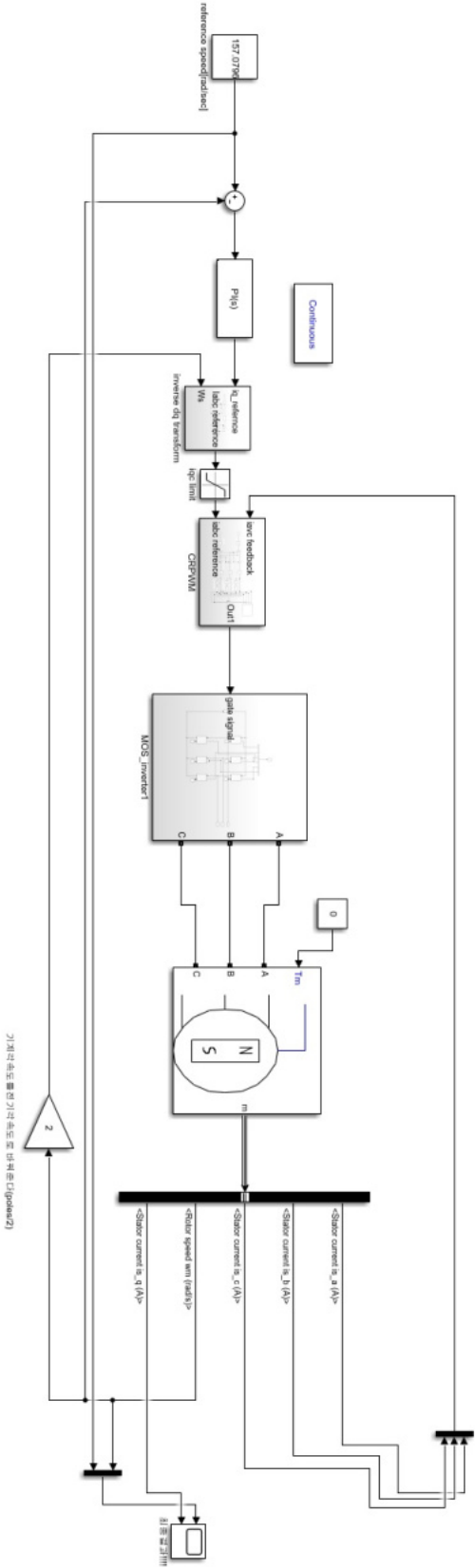
[그림26] reference speed and settling time

Reference speed값이 157.1[rad/sec]이고 Rotor speed값이 153.9[rad/sec]이다. Rotor speed가 Reference 2% 오차 안에 들어가므로 설계 조건을 만족 한다고 볼 수 있다. 그리고 4-(1)에서 보았듯이 stator  $I_q$  PI controller 신호를 따라 간다는 점을 볼 수 있다. 추가로 다음 사진은  $T_s$  부근을 확대한 사진이다.



[그림27]  $T_s$  부근

다음 그림 최종 설계한 모습이다.



[그림28] SIMULINK 모습

## 5. 고찰/참고

### • 5-A 고찰

지금 PMSM(Permanent Magnet Synchronous Motor)의 속도를 제어할 수 있는 controller를 설계했다. 설계의 목적은 단순히 속도를 제어하는 것이 아니라 Reference speed를 입력했을 때 안정적으로 속도를 제어하는 것이다.

단상을 다상으로 바꾸기 위해 Inverse-dq0를 사용했고, 전류를 일정하게 제어하기 위해 CRPWM을 사용했다. 그리고 PMSM 제어기의 하드웨어인 MOSFET Inverter를 이용해 3상전류를 제어했다. 마지막으로 PI controller를 통해 Reference speed를 원하는 파형으로 제어할 수 있었다.

2-(1)에서 말했듯이 PMSM은 일상생활에서 쓰이는 작은 모터부터 산업용으로 쓰이기까지 범위가 다양하다. 그만큼 정밀도가 요구된다고 볼 수 있다. 이것을 만족하기 위해서 Rising time은 작지만 그만큼 Overshoot도 설정했다. 그리고 비록 preset model 1번이긴 했지만 전류가 크게 흐르면 모터입장에서도 부담이 크므로 낮은 값으로 수렴하는 것을 알 수 있었다.

이번 수업 때 SIMULINK를 처음 사용했는데 정말 편하고 유용하다는 것을 느꼈다. PMSM을 제어하는 방법도 배웠지만 SIMULINK을 다루는 방법을 배웠다는 것에 큰 의미가 있다고 느껴진다. 복잡한 코딩이 아니라 한 눈에 알아볼 수 있는 Block을 사용하고 교제에서 배운 대로 선을 연결해 시뮬레이션을 돌린다는 점이 아주 효율적이라고 생각한다.

### 5-B Reference

- 1) <https://functionbay.com> PMSM - 2page
- 2) Stephen D. Umans 『전기기기 7<sup>th</sup> Edition』 349p - 3page
- 3) [http://www.ktword.co.kr/test/view/view.php?nav=2&id=922&m\\_temp1=5976](http://www.ktword.co.kr/test/view/view.php?nav=2&id=922&m_temp1=5976) - 4page
- 4) Stephen D. Umans, 『전기기기 7<sup>th</sup> Edition』, 235p - 5page
- 5) Richard C. Dorf, 『Modern Control Systems』, 416p - 5page
- 6) Stephen D. Umans, 『전기기기 7<sup>th</sup> Edition』, 739p - 5page
- 7) <http://infoupdate.org/synchronous-motor-stator/> - 6page
- 8) <https://kr.mathworks.com/help/physmod/sps/ref/parktransform.html> - 6page
- 9) [https://www.researchgate.net/figure/Diagram-of-hysteresis-current-control\\_fig3\\_277131487](https://www.researchgate.net/figure/Diagram-of-hysteresis-current-control_fig3_277131487) - 7page
- 10) <http://mwb-online.co/cissell-wiring-diagrams/> - 8page