Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

강사 – **Innova Lee(이상훈)**
gcccompil3r@gmail.com
학생 – 변진혁
xollgun@gmail.com
학생 – 김형준
kimdj417@gmail.com

BattleShip
6월 2주차

위대한 수학자 일개미: 김형준
쥐뿔도 모르는 베짱이: 변진혁

6월 2주차(7일 ~ 14일) 진행사항

변진혁

- I2C통신을 이용한 센서 사용 연동

- ADC와 MCU 연동

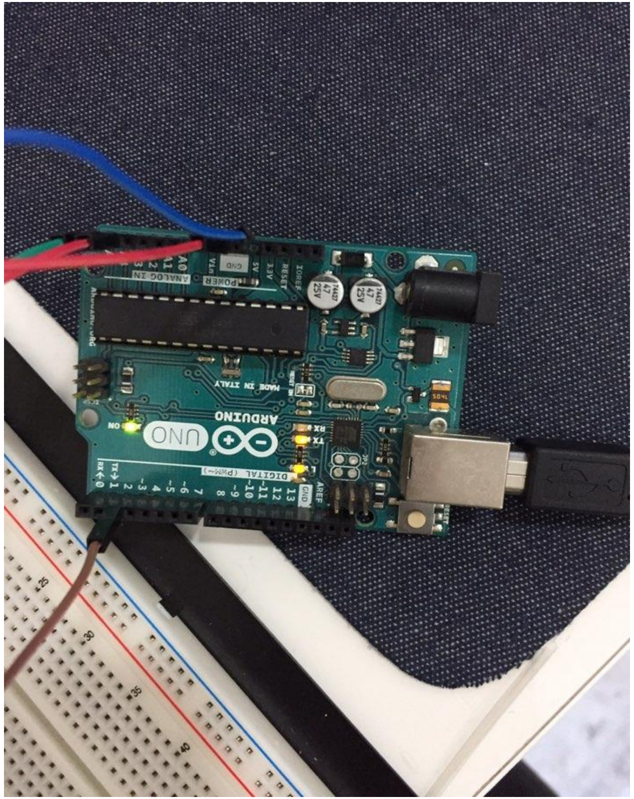
- MPU9250 선정

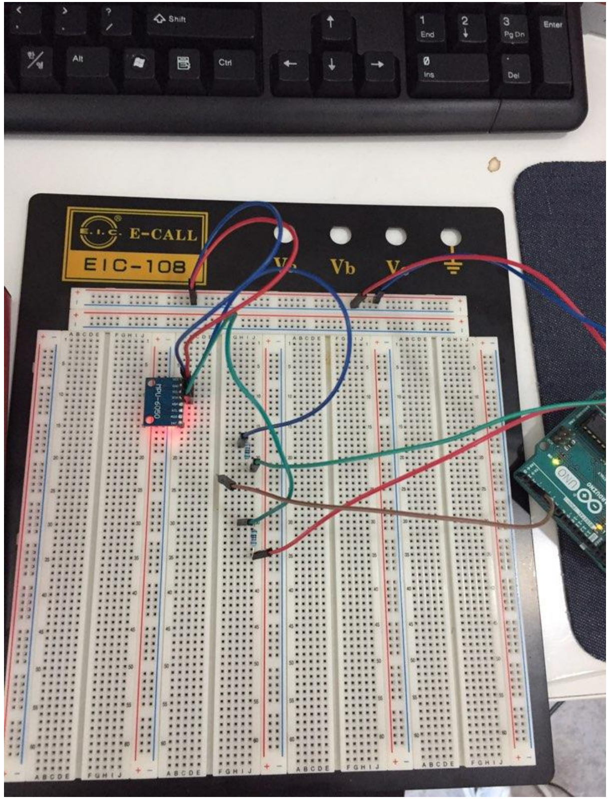
MPU 선정 완료

MPU - 9250

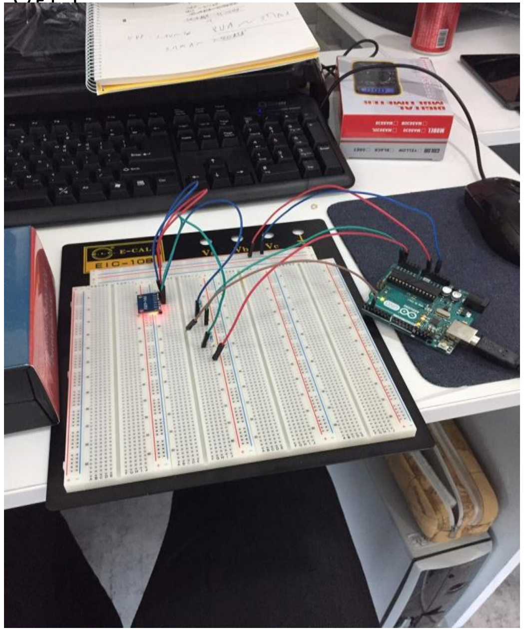
MPU를 선정 하였지만 문제점 발생
MPU의 최대 가용할수 있는 A(암페어)에 대한 정보가 없다…. 해당 제조사에 문의를 해야되는 상황….. 제조사(Invensense)는 미국기업이다…. 악~~~!!!!
영어로 문의해야되는 상황….
영알못인데 영어로 문의를 해야된다.

MPU6050 -> Arduino I2C TEST(1)

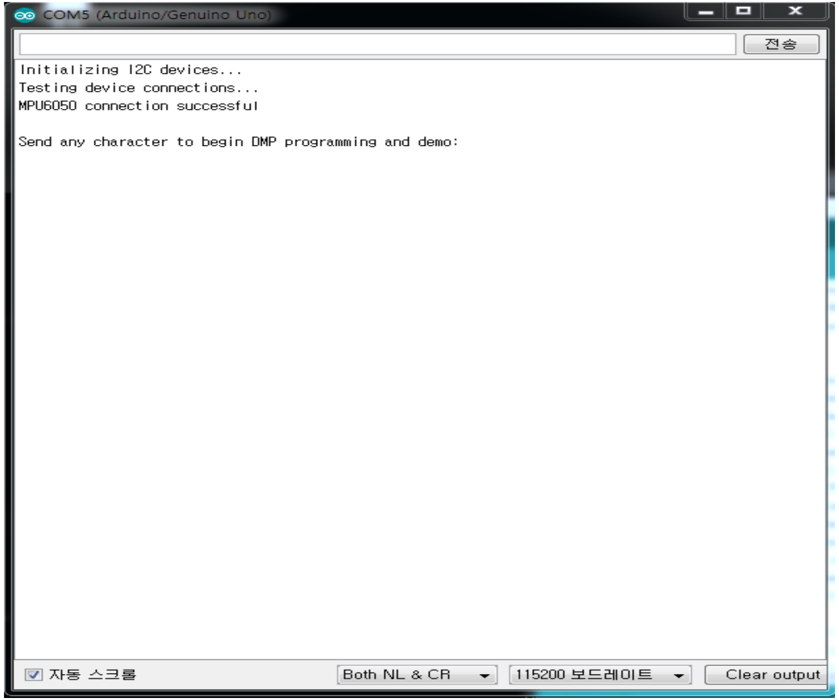




MPU와 MCU를 연결하기 전에
먼저 아두이노 TEST를 해보았다!

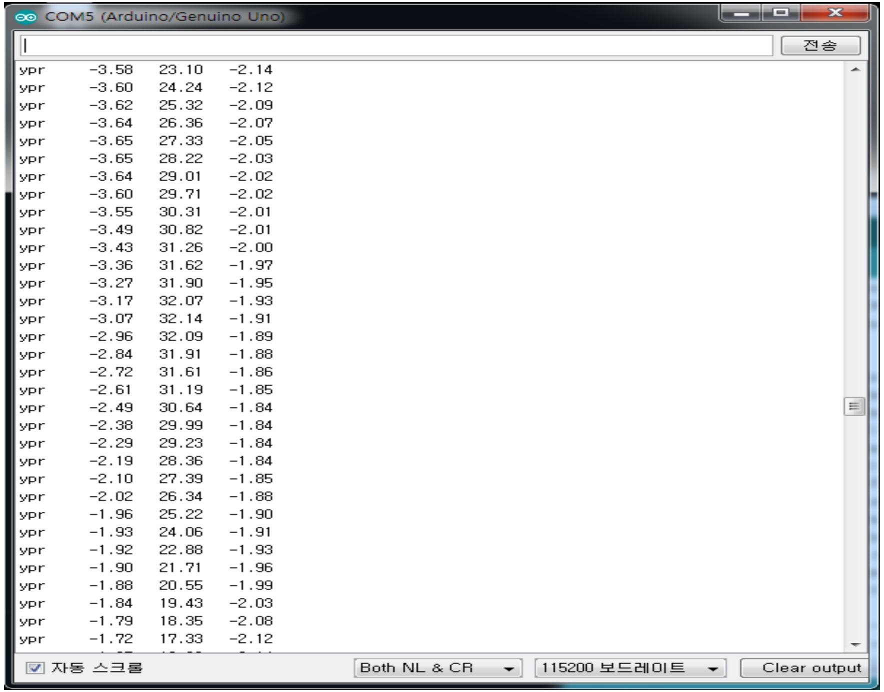


MPU6050 -> Arduino I2C TEST(2)

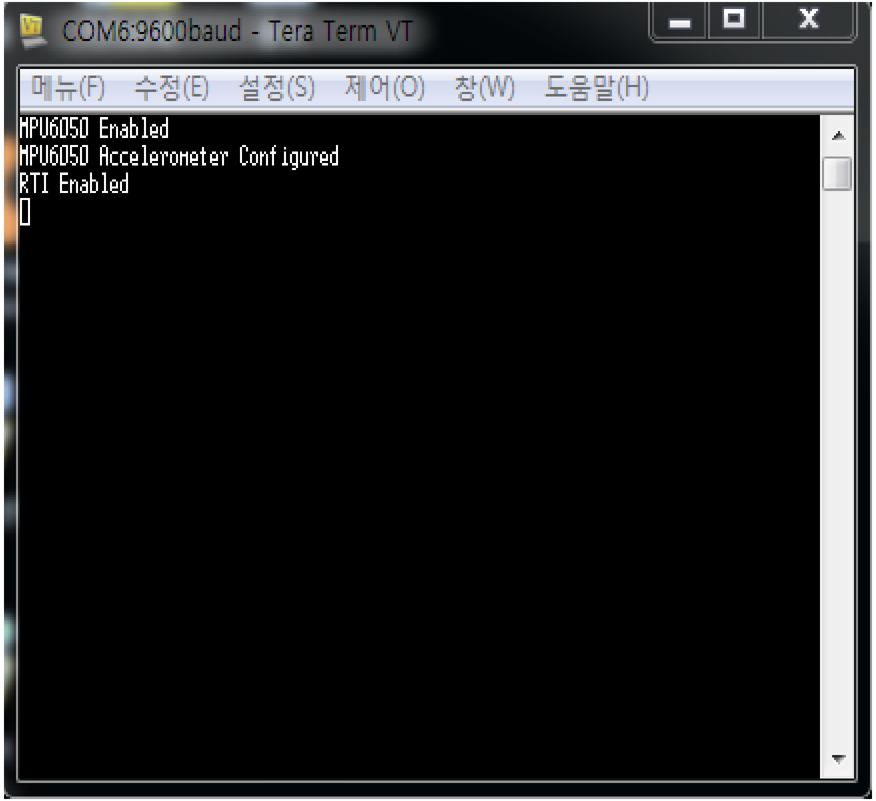


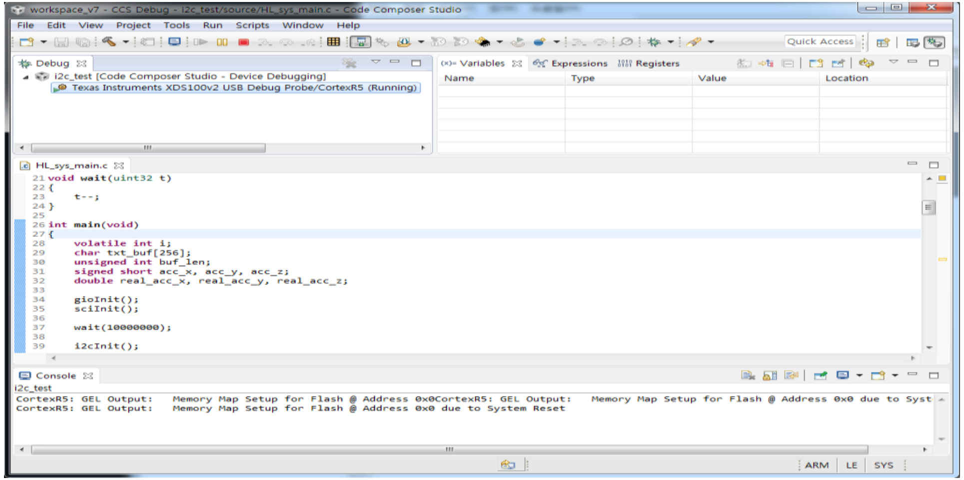
MPU6050과 연결되었다는 메시지

수치 값이 잘 나온다. yaw pitch raw 순서



MPU6050 -> MCU I2C TEST





신호가 불안정 해서 그런지는 모르겠지만 MPU와 연결 되었다는 메세지는 있지만 데이터 값이 나오지 않았다.

6월 2주차(7일 ~ 14일) 진행사항

김형준

ESC배터리 구매

BLDC 데이터시트

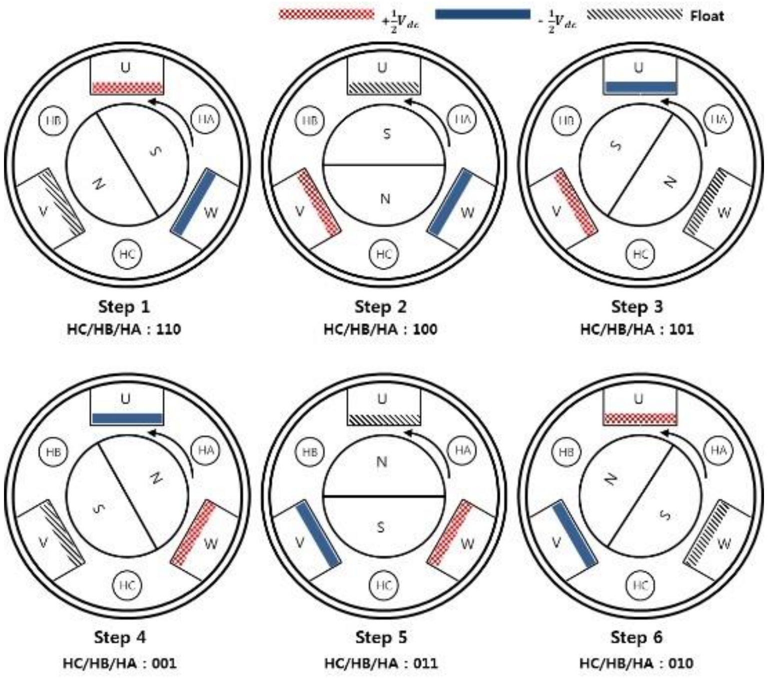
PI controller(MATLAB & SIMULINK) 구현

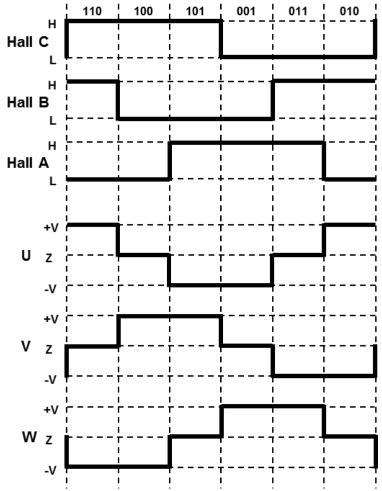
PI controller ccs 구현중

ESC 배터리 구매

- Size(크기, L*W*H) : 138*46*49mm
- Voltage: 14.8v
- Amp : 7400mA
- cell : 4s
- Weight(무게) : 580g(w/Deans connector)
- Charge Current(충전전류) : **Max 5C**
- C-Rate(방전율) : **55C**
- M-Rate(방전율) :**120C**

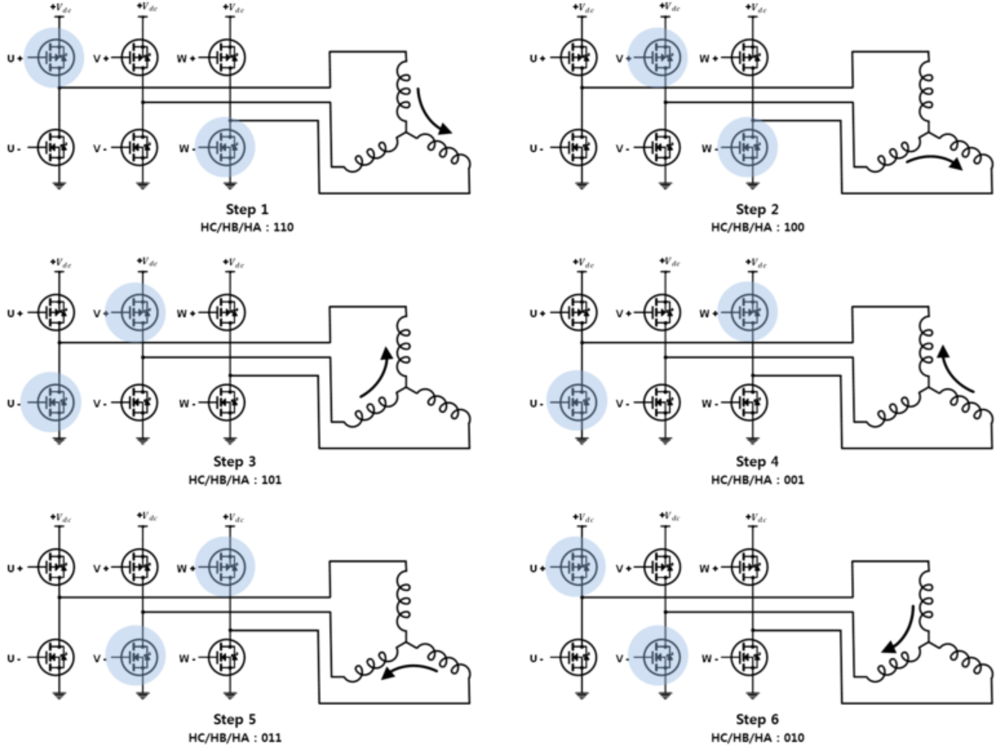
BLDC 구동 원리





첫 번째 step 부터 살펴보면 회전자가 반시계 방향으로 회전하기 위해서 고정자 U, V, W 상은 각각 어떤 자속을 발생시켜야 할지 보겠습니다. U 상은 회전자의 S 극을 당겨야 하므로 N 극이 될 수 있게 전류를 흘려야 하고, W 상은 회전자의 S 극을 밀어야 하므로 S 극이 될 수 있게 전류를 흘려야 합니다. V 상은 애매한 위치에 있으므로 float 상태로 전류를 흘려보내지 않으면 됩니다. 이때의 홀센서 신호를 살펴보겠습니다. 홀센서가 자석의 S 극이 다가올 때는 LOW (0), N 극이 다가올 때는 HIGH (1) 라고 가정하면 HA는 S 극이 가까이에 있으므로 0, HB와 HC는 N 극이 가까이에 있으므로 1이 됩니다. 따라서 HC/HB/HA 의 신호 조합은 110이라 할 수 있겠습니다.

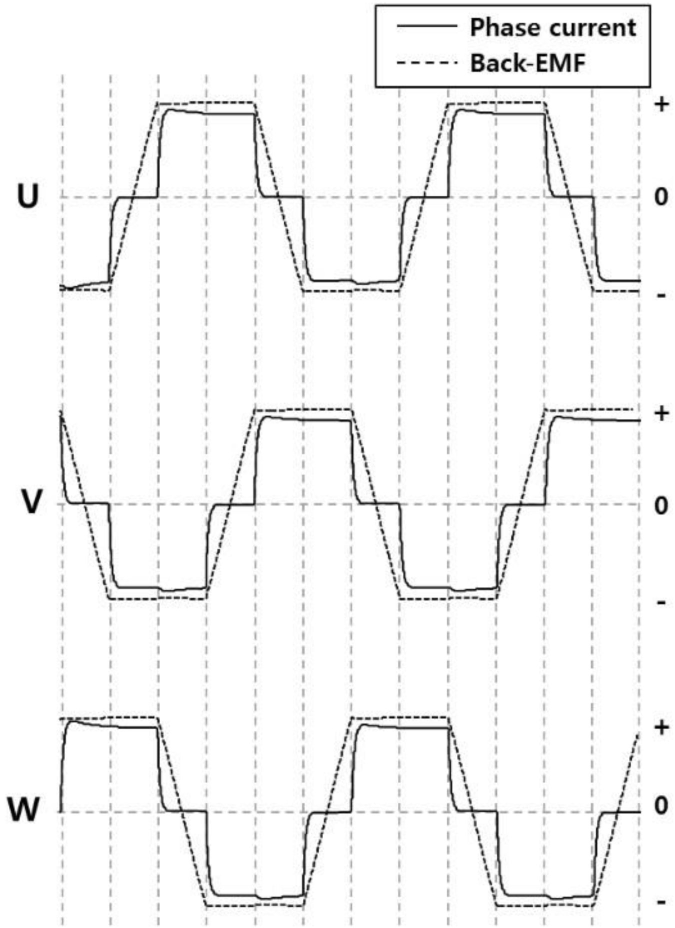
[출처] [2] BLDC 모터의 구동 원리|작성자 엠에스리



	Hall sensor			Switching driver					
	HC	HB	HA	U+	U-	V+	V-	W+	W-
1	1	1	0	1	0	0	0	0	1
2	1	0	0	0	0	1	0	0	1
3	1	0	1	0	1	1	0	0	0
4	0	0	1	0	1	0	0	1	0
5	0	1	1	0	0	0	1	1	0
6	0	1	0	1	0	0	1	0	0

권선 3개가 각각 U 상, V 상, W 상에 해당하는 권선입니다. 그리고 각 권선마다 MOSFET (스위칭 소자)가 2개씩 총 6개로 구성되어있습니다. 위쪽에 MOSFET 3개가 각 상마다 +방향으로 전류를 흘려주게 하는 역할을 하고, 아래쪽 MOSFET 3개가 각 상마다 -방향으로 전류를 흘려주게 합니다.

[출처] [2] BLDC 모터의 구동 원리|작성자 엠에스리

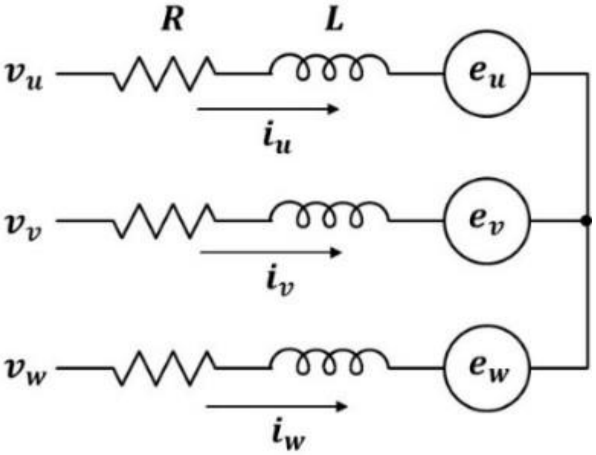


	BLDC motor	PMSM
Back-EMF	Trapezoidal	Sinusoidal
Phase current	Quasi-square waveform	Sinusoidal
Driving method	- Two-phase excitation method - Rotor position is detected by hall sensors	- Torque control in vector control - High resolution position sensor is needed
Torque ripple	Moderate	Low
Cost	Low	High

$$\begin{bmatrix} v_u \\ v_v \\ v_w \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{bmatrix} \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} + \begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix}$$

$$\begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix} = \omega_m \lambda_m \begin{bmatrix} f_u(\theta_e) \\ f_v(\theta_e) \\ f_w(\theta_e) \end{bmatrix} = k_e \omega_m$$

λ_m : 자속
ω_m : 각가속도
θ_e : 회전자 위치회전자 위치
$f(\theta_e)$: 사다리꼴 방정식
k_e : 역기전력 상수



$v_■$: 입력전압
$i_■$: 상전류
$e_■$: 역기전력

R : 저항
L : 인덕턴스
M : 상호 인덕턴스

$$f(\theta_e) = \begin{cases} 6\theta_e/\pi & 0 \le \theta_e < \pi/6 \\ 1 & \pi/6 \le \theta_e < 5\pi/6 \\ -6\theta_e/\pi + 6 & 5\pi/6 \le \theta_e < 7\pi/6 \\ -1 & 7\pi/6 \le \theta_e < 11\pi/6 \\ 6\theta_e/\pi - 12 & 11\pi/6 \le \theta_e < 2\pi \end{cases}$$

$$\tau_e = k_T I$$

$$\tau_e = \frac{e_u i_u + e_v i_v + e_w i_w}{\omega_m}$$

τ_L : 부하 토크
τ_e : 전동기의 토크
k_T : 토크 상수 [Nm/A]
ω_m : 각속도

$$P = V \times I = \tau_e \times \omega_m$$

[W]=[V]·[A]=[Nm]·[/s]

$$k_T = [\mathrm{Nm}]/[\mathrm{A}] \qquad k_e = [\mathrm{V}] \cdot [\mathrm{s}]$$

$$V = E + IR = k_e \omega_m + IR$$
$$VI = EI + I^2 R = k_e \omega_m I + I^2 R$$

$$\tau_e \omega_m = k_T I \omega_m = k_e \omega_m I$$

$$J \frac{d\omega_m}{dt} + B\,\omega_m = (\tau_e - \tau_L)$$

J : 관성 모멘트

B : 댐핑 계수

PID Controller

P(proportionally), I(Integral), D(derivative) 비례 적분 미분 제어기

기본적으로 피드백(feedback)제어기의 형태를 가지고 있으며, 제어하고자 하는 대상의 출력값(output)을 측정하여 이를 원하고자 하는 참조값(reference value) 혹은 설정값(setpoint)과 비교하여 오차(error)를 계산하고, 이 오차값을 이용하여 제어에 필요한 제어값을 계산하는 구조로 되어 있다.

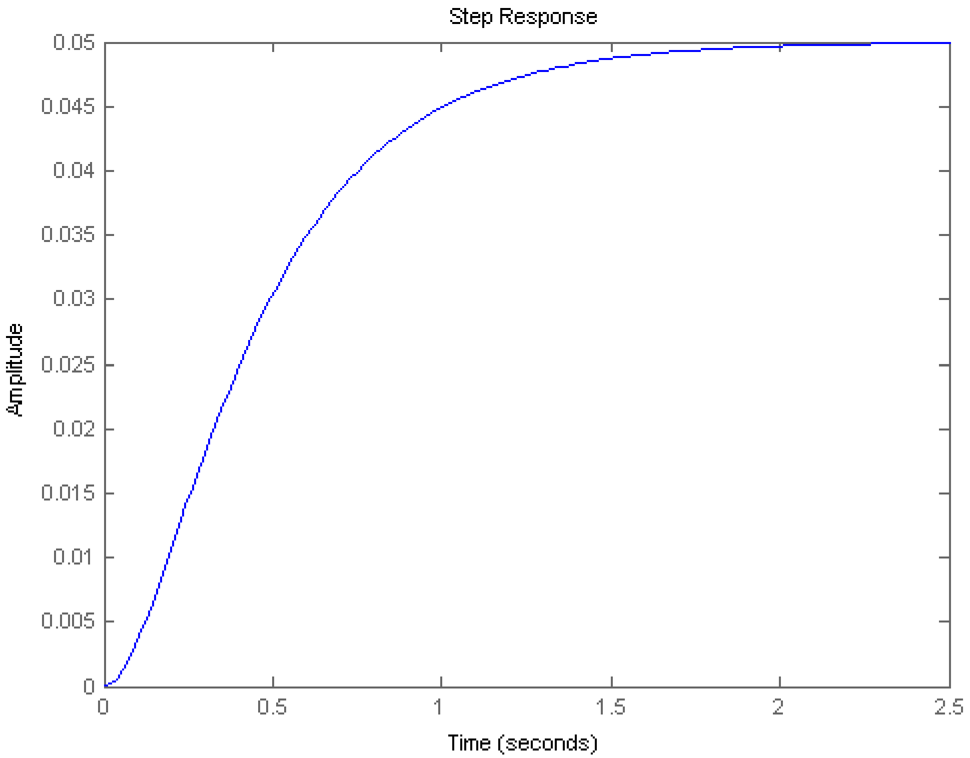
$$u(t) = K_p e(t) + K_i \int e(t)dt + K_p \frac{de}{dt}$$

Laplace Transform

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

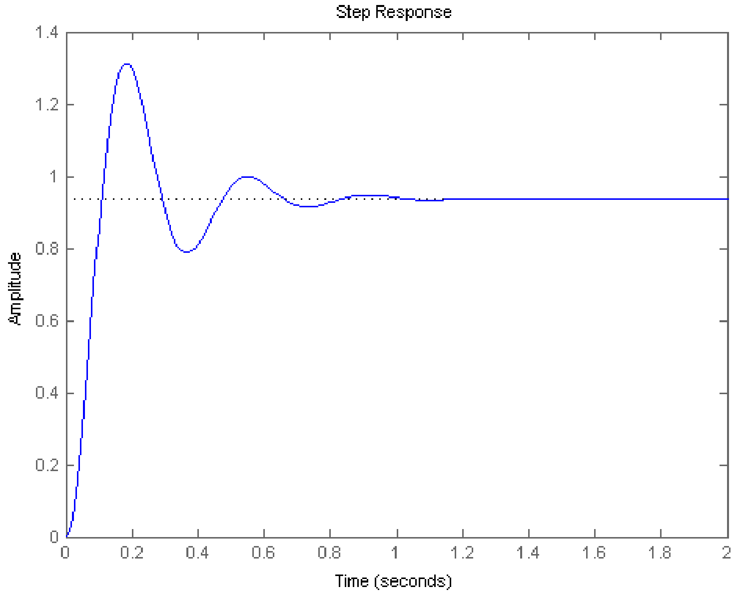
Open loop Response

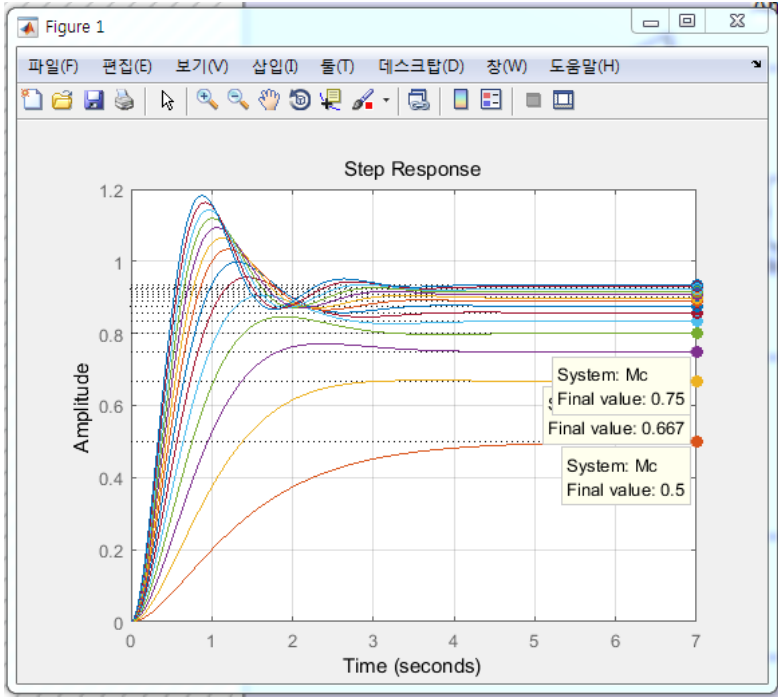
$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 10s + 20}$$



Proportional Control

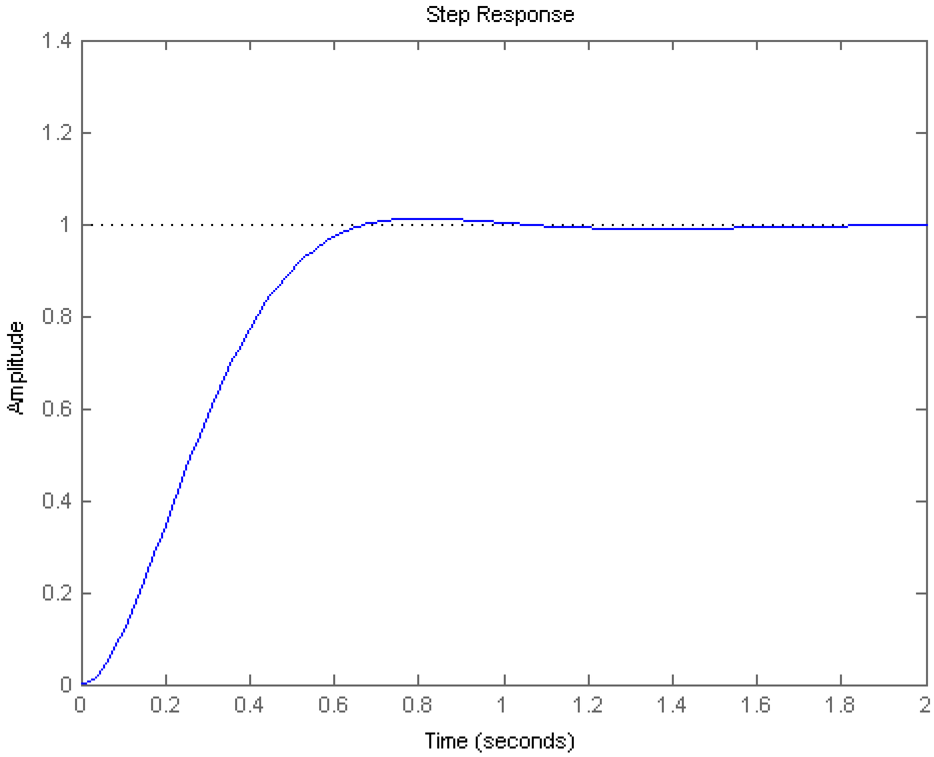
$$\frac{X(s)}{F(s)} = \frac{K_p}{s^2 + 10s + (20 + K_p)}$$





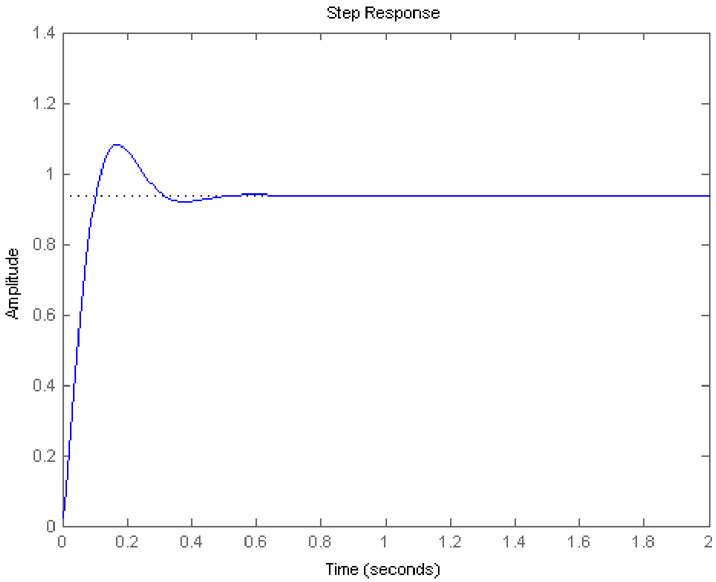
proportional-Integral

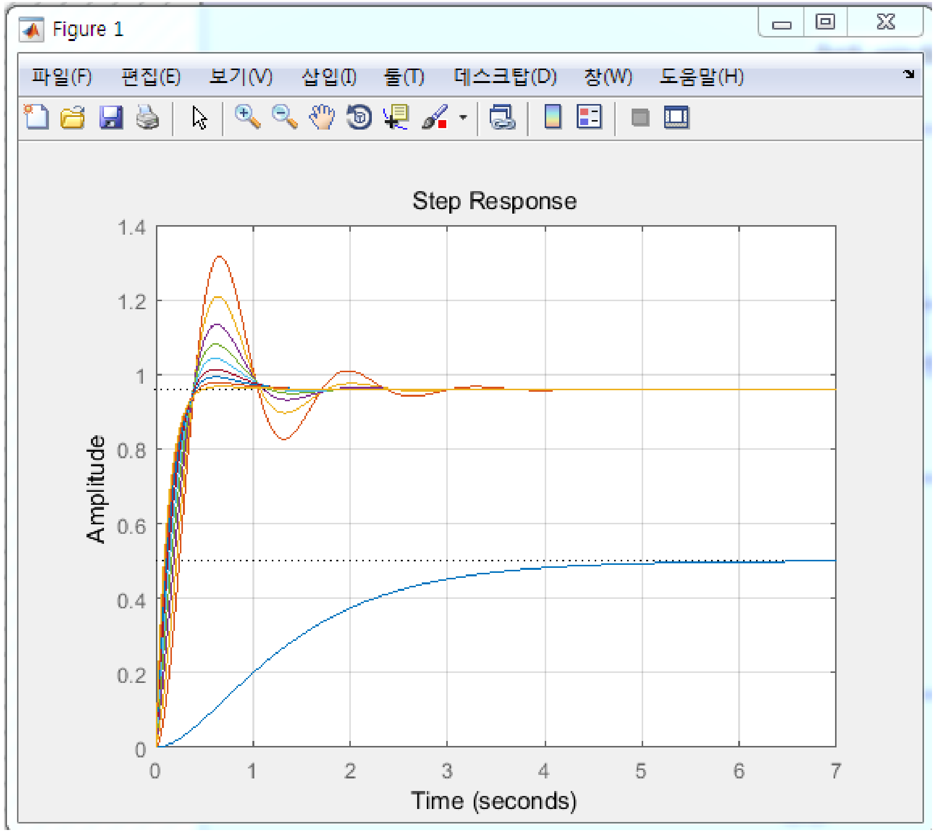
$$\frac{X(s)}{F(s)} = \frac{K_p s + K_i}{s^3 + 10s^2 + (20 + K_p s + K_i)}$$



Proportional-Derivative

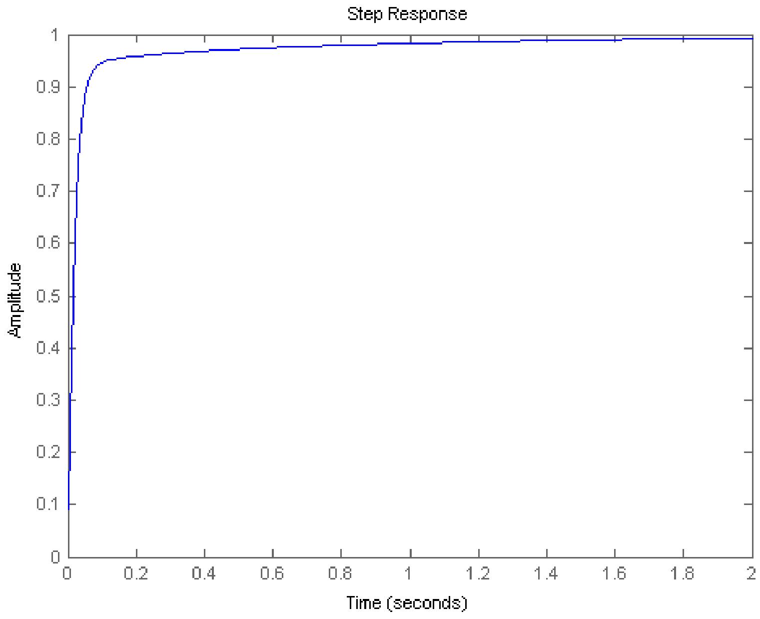
$$\frac{X(s)}{F(s)} = \frac{K_d s + K_p}{s^2 + (10 + K_d)s + (20 + K_p)}$$

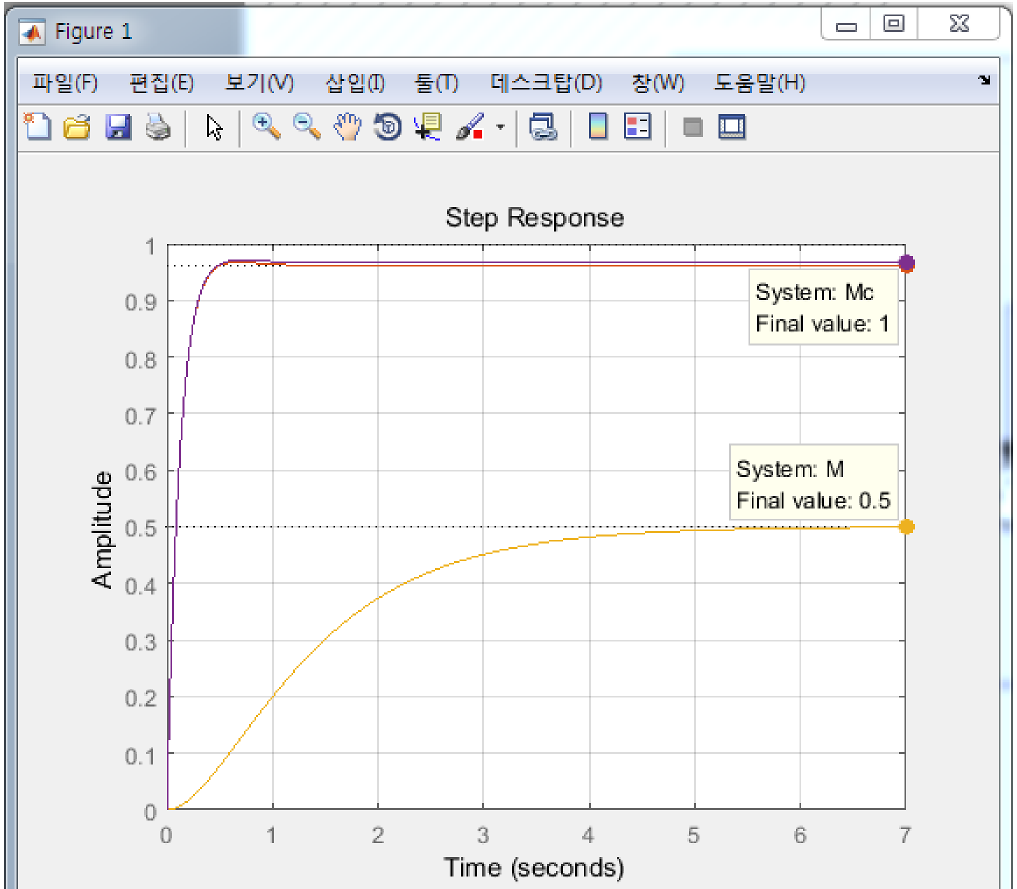




Proportional-Integral-Derivative

$$\frac{X(s)}{F(s)} = \frac{K_d s^2 + K_p s + K_i}{s^3 + (10 + K_d)s^2 + (20 + K_p)s + K_i}$$





Matlab Calculation

```
nom =

     1


denom =

     1     3     1


Gp =

        1
  -------------
  s^2 + 3 s + 1

Continuous-time transfer function.


H =

     1


M =

        1
  -------------
  s^2 + 3 s + 2

Continuous-time transfer function.
```

```
Kp =

    24


Ki =

     1


Kd =

     8


Gc =

             1
  Kp + Ki * --- + Kd * s
             s

  with Kp = 24, Ki = 1, Kd = 8

Continuous-time PID controller in parallel form.


Mc =

     8 s^2 + 24 s + 1
  -----------------------
  s^3 + 11 s^2 + 25 s + 1

Continuous-time transfer function.
```

PID의 작용

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small Change	Decrease	Decrease	No Change

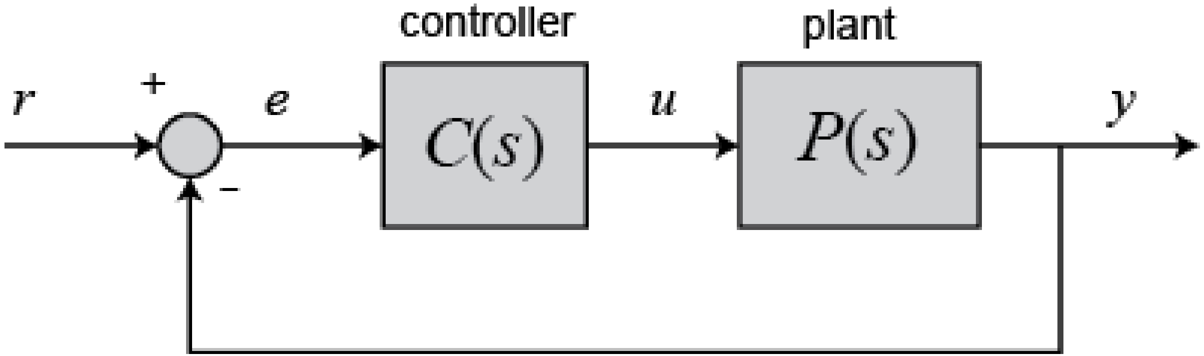
General Tips for Designing a PID Controller

When you are designing a PID controller for a given system, follow the steps shown below to obtain a desired response.

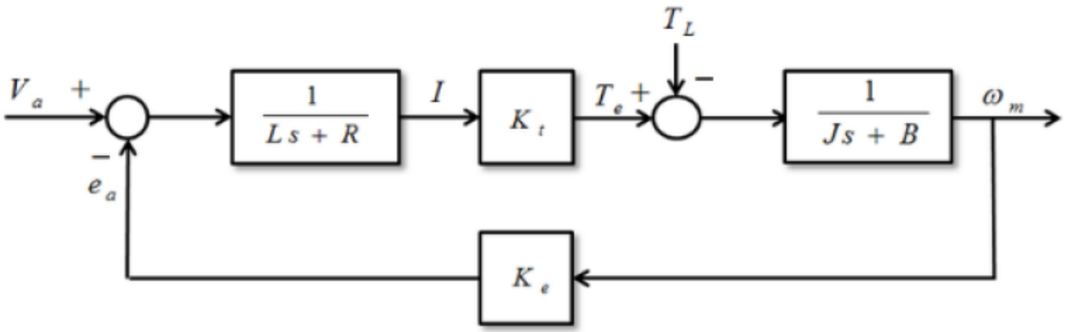
1. Obtain an open-loop response and determine what needs to be improved
2. Add a proportional control to improve the rise time
3. Add a derivative control to improve the overshoot
4. Add an integral control to eliminate the steady-state error
5. Adjust each of Kp, Ki, and Kd until you obtain a desired overall response. You can always refer to the table shown in this "PID Tutorial" page to find out which controller controls what characteristics.

Lastly, please keep in mind that you do not need to implement all three controllers (proportional, derivative, and integral) into a single system, if not necessary. For example, if a PI controller gives a good enough response (like the above example), then you don't need to implement a derivative controller on the system. Keep the controller as simple as possible.

BLDC 모터 제어 방식



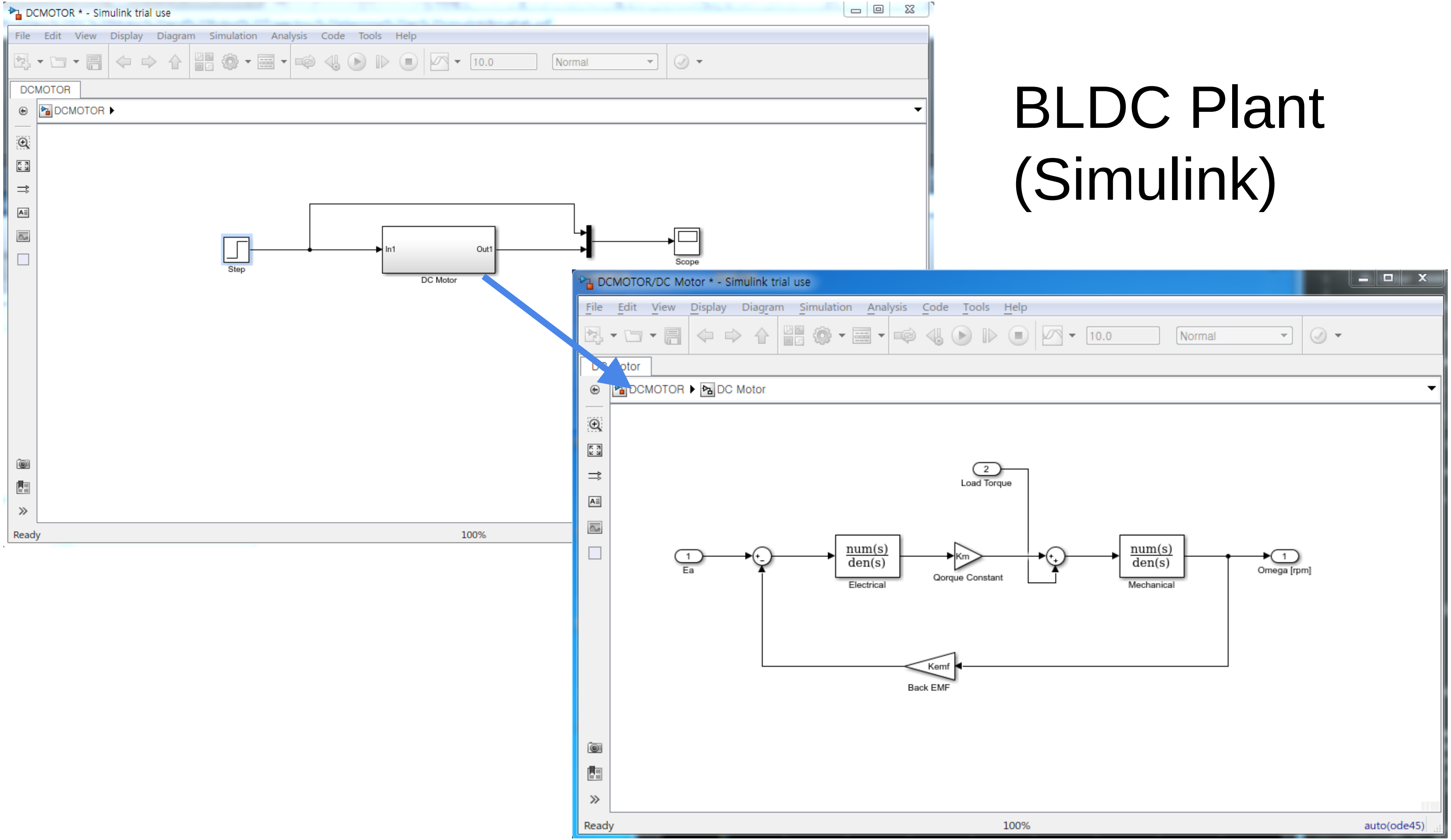
BLDC 모터 모델링(plant)



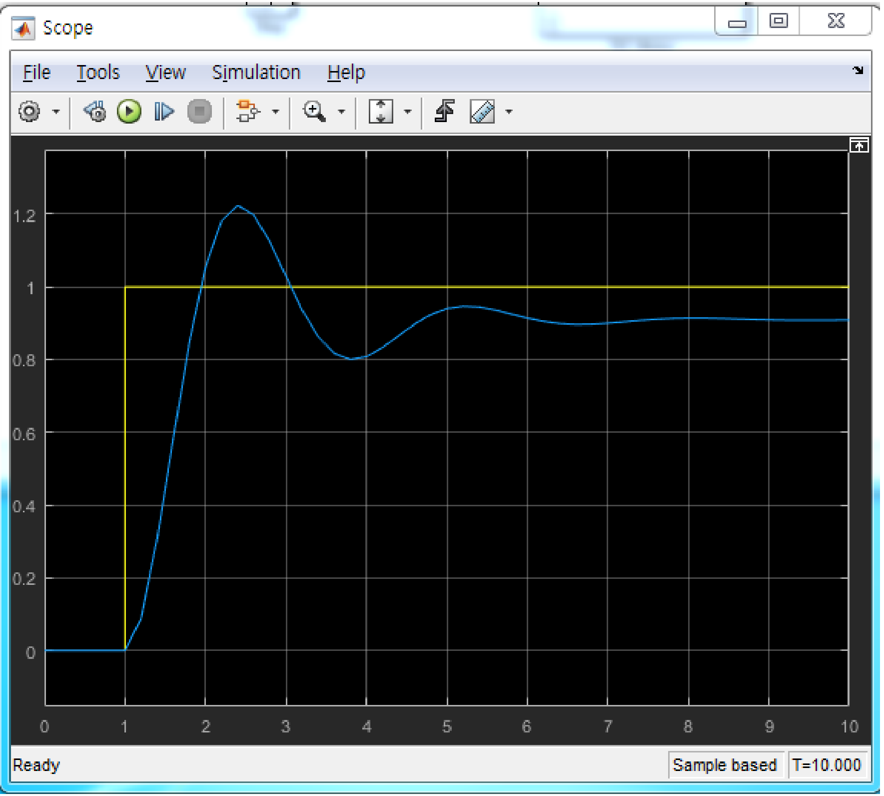
DC모터의 구동시스템 블록도

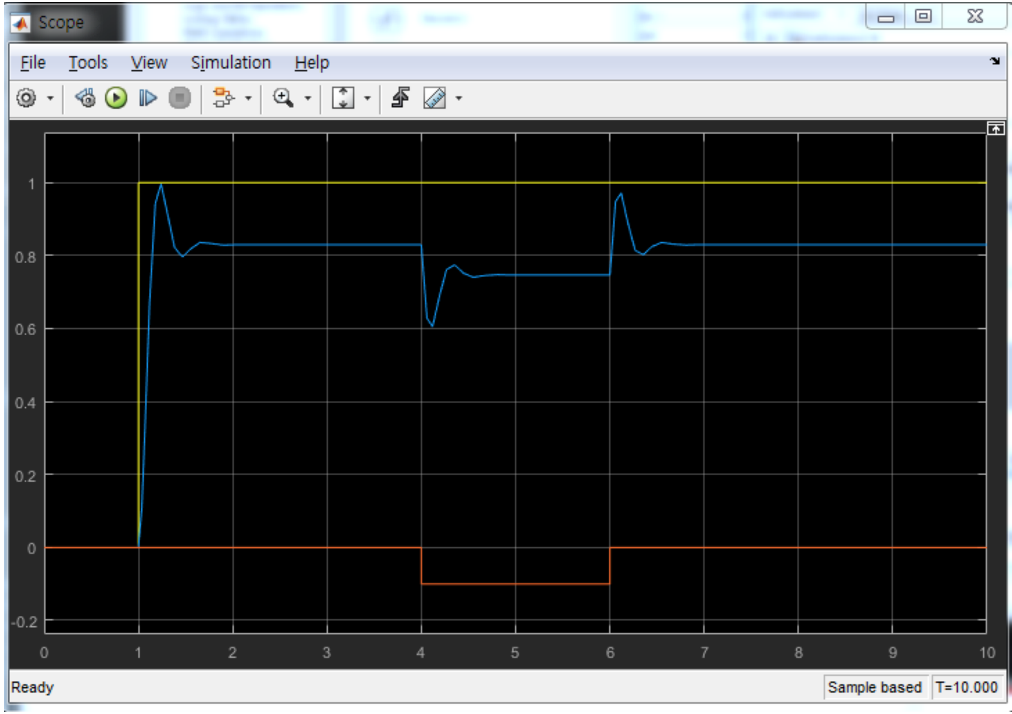
$$\frac{\omega_m(s)}{v(s)} = \frac{k_T}{(L_a s + R_a)(Js + B) + k_T k_e}$$

BLDC Plant (Simulink)

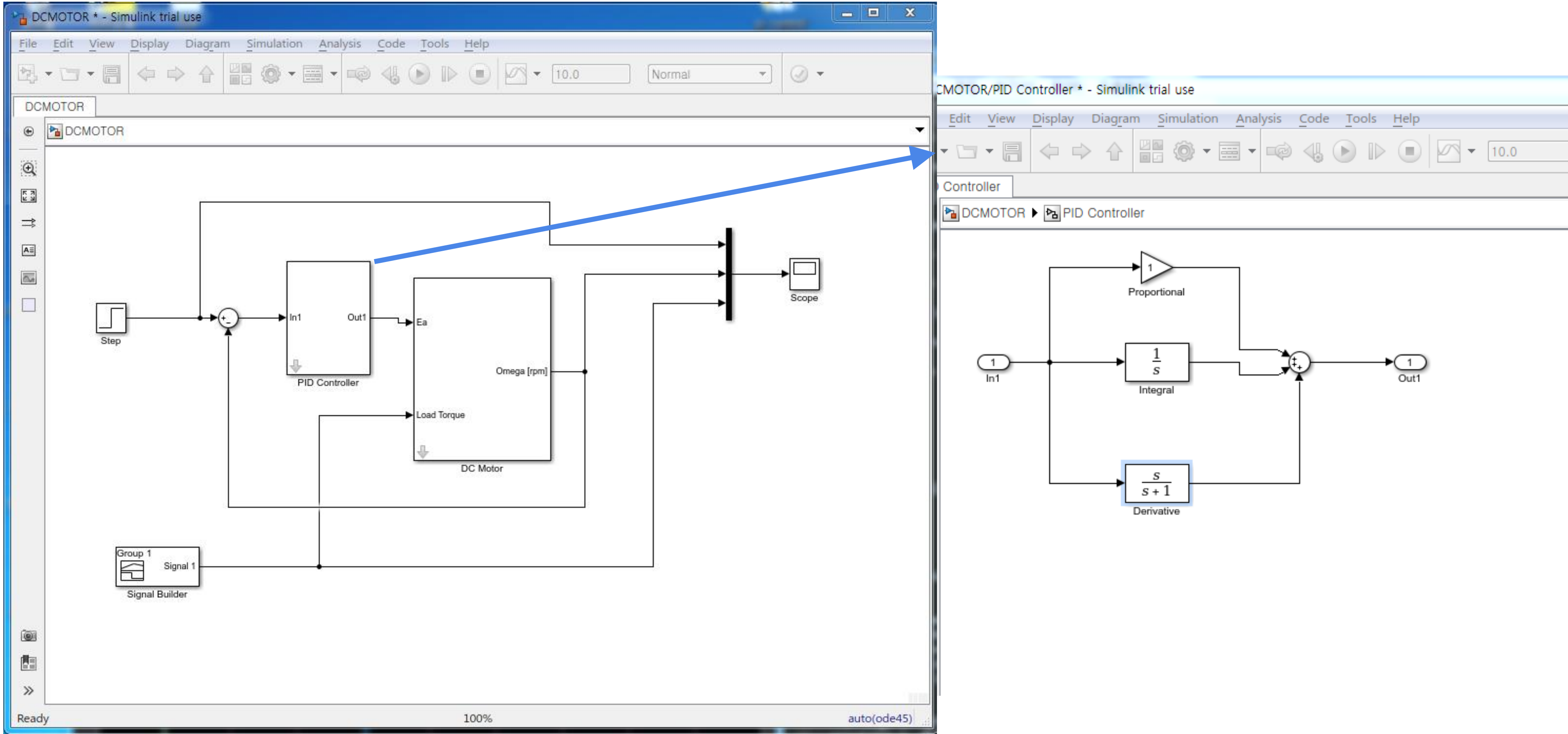


BLDC Plant scope

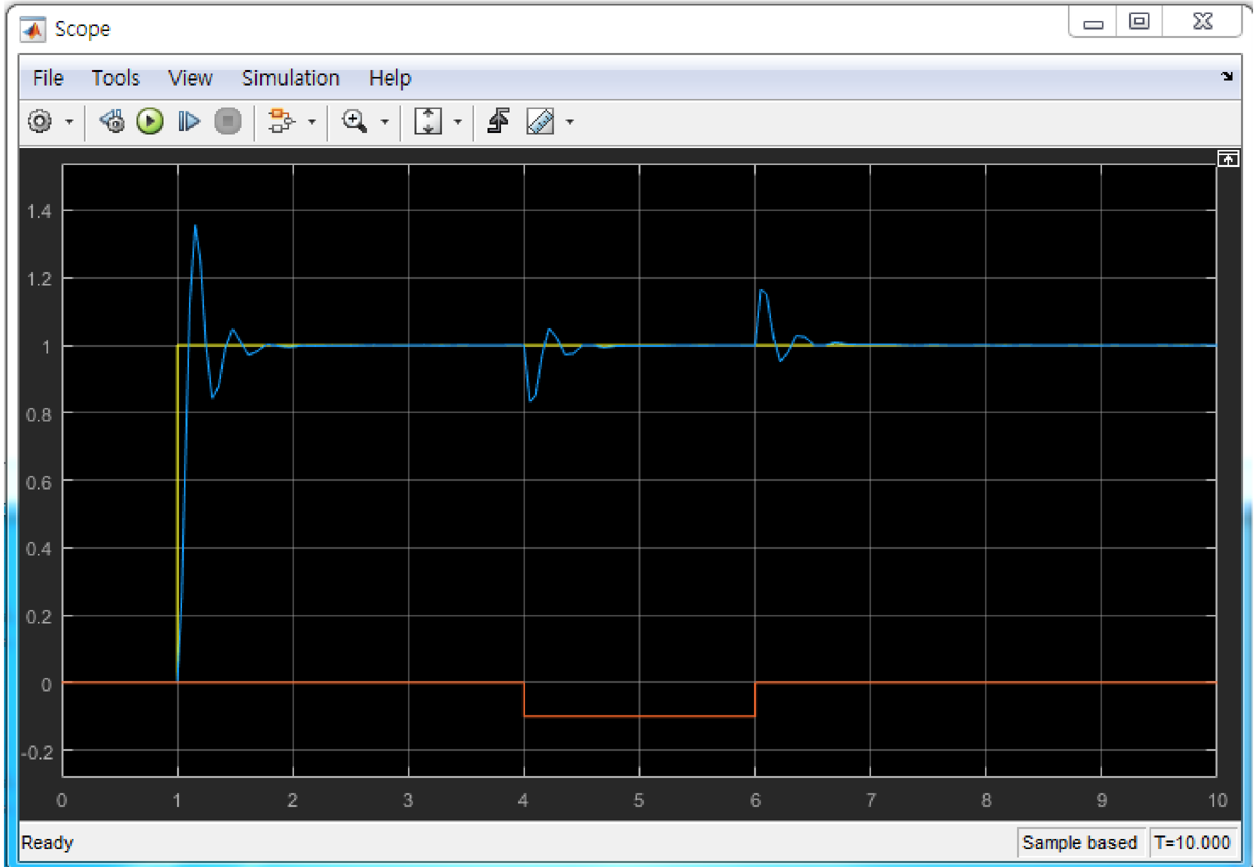




Simulink PID Controller + Plant



PI Controller (+ 부하 발생 상황)



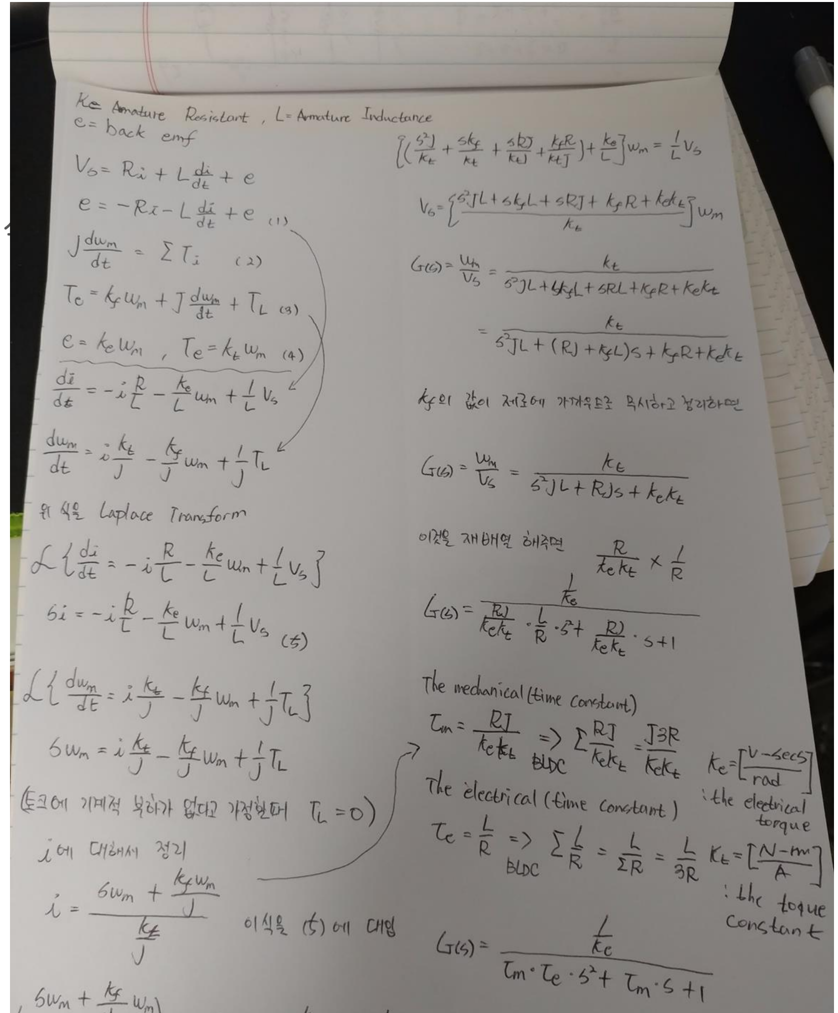
SSS Motor Data & Characteristics

	SSS Motor Data	Unit	value
	values at nominal voltage		
1	Nominal Voltage	v	37
2	No load speed	rpm	35000
3	No load Current	A	45
4	Nominal Speed	rpm	25000
5	Nominal Torque	mNm	59
6	Nominal Current	A	2.14
7	Stall Torque	mNm	255
8	Starting Current	A	10.0
9	Maximum Efficiency	%	90

	characteristics		
10	Terminal Resistance phase to phase	amp	1.2
11	Terminal Inductance phase to phase	mH	0.560
12	Torque Constant	mNm/A	25.5
13	speed Constant	rpm/V	37.4
14	speed/Torque Gradient	rpm/mNm	17.6
15	Mechanical time constant	ms	17.1
16	Rotor Inertia	gcm^2	92.5
17	number of phase		3

Plant Modeling

모터의 전압과 각속도의 관계



R = Armature Resistant , L = Armature Inductance
e = back emf

$V_s = Ri + L\frac{di}{dt} + e$

$e = -Ri - L\frac{di}{dt} + e$ (1)

$J\frac{dw_m}{dt} = \sum T_i$ (2)

$T_e = k_f w_m + J\frac{dw_m}{dt} + T_L$ (3)

$e = k_e w_m$, $T_e = k_t w_m$ (4)

$\frac{di}{dt} = -i\frac{R}{L} - \frac{k_e}{L}w_m + \frac{1}{L}V_s$

$\frac{dw_m}{dt} = i\frac{k_t}{J} - \frac{k_f}{J}w_m + \frac{1}{J}T_L$

위 식을 Laplace Transform

$\mathcal{L}\{\frac{di}{dt} = -i\frac{R}{L} - \frac{k_e}{L}w_m + \frac{1}{L}V_s\}$

$si = -i\frac{R}{L} - \frac{k_e}{L}w_m + \frac{1}{L}V_s$ (5)

$\mathcal{L}\{\frac{dw_m}{dt} = i\frac{k_t}{J} - \frac{k_f}{J}w_m + \frac{1}{J}T_L\}$

$sw_m = i\frac{k_t}{J} - \frac{k_f}{J}w_m + \frac{1}{J}T_L$

(토크에 기계적 부하가 없다고 가정할때 $T_L = 0$)

i에 대해서 정리

$i = \frac{sw_m + \frac{k_f w_m}{J}}{\frac{k_t}{J}}$ 이식을 (5)에 대입

$sw_m + \frac{k_f}{J}w_m)$

$[(\frac{s^2J}{k_t} + \frac{sk_f}{k_t} + \frac{sRJ}{k_tJ} + \frac{k_fR}{k_tJ}) + \frac{k_e}{L}]w_m = \frac{1}{L}V_s$

$V_s = [\frac{s^2JL + sk_fL + sRJ + k_fR + k_ek_t}{k_t}]w_m$

$G(s) = \frac{w_m}{V_s} = \frac{k_t}{s^2JL + sk_fL + sRL + k_fR + k_ek_t}$

$= \frac{k_t}{s^2JL + (RJ + k_fL)s + k_fR + k_ek_t}$

k_f의 값이 제로에 가까우므로 무시하고 정리하면

$G(s) = \frac{w_m}{V_s} = \frac{k_t}{s^2JL + RJs + k_ek_t}$

이것을 재배열 해주면 $\frac{R}{k_ek_t} \times \frac{1}{R}$

$G(s) = \frac{\frac{1}{k_e}}{\frac{RJ}{k_ek_t}\cdot\frac{L}{R}\cdot s^2 + \frac{RJ}{k_ek_t}\cdot s + 1}$

The mechanical (time constant)

$\tau_m = \frac{RJ}{k_ek_t} \Rightarrow \sum_{BLDC} \frac{RJ}{k_ek_t} = \frac{J3R}{k_ek_t}$

$k_e = [\frac{V-secs}{rad}]$: the electrical torque

The electrical (time constant)

$\tau_e = \frac{L}{R} \Rightarrow \sum_{BLDC} \frac{L}{R} = \frac{L}{\Sigma R} = \frac{L}{3R}$

$k_t = [\frac{N-m}{A}]$: the toque constant

$G(s) = \frac{\frac{1}{k_e}}{\tau_m\cdot\tau_e\cdot s^2 + \tau_m\cdot s + 1}$

555 3660 Motor 에 속도와 각속도의
관계식 적용

$$\tau_e = \frac{0.560 \times 10^{-3}}{3 \times 1.20} = \frac{L}{3R}$$

$$= 155.56 \times 10^{-6}$$

$$\tau_m = \frac{3RJ}{k_e \cdot k_t}$$

$R = 1.2$

$J_{Rotor} = 92.5\ gcm^2 = 9.25 \times 10^{-6}\ kgm^2$

$k_t = 25.5 \times 10^{-3}\ Nm/A$

$\tau_m = 0.0171\ secs$

$$\tau_m = \frac{3RJ}{k_e k_t} = 0.0171$$

$$k_e = \frac{3RJ}{k_t \cdot \tau_m} = \frac{3 \times 1.2 \times 9.25 \times 10^{-6}}{0.0171 \times 25.5 \times 10^{-3}} = 0.0763\ \frac{V\text{-}secs}{rad}$$

$$G(s) = \frac{13.11}{155.56 \times 10^{-6} \times 0.0171 \cdot s^2 + 0.0171 s + 1}$$

$$= \frac{13.11}{2.66 \times 10^{-6} \cdot s^2 + 0.0171 \cdot s + 1}$$

R = Armature Resistance

L = Armature Inductance

V_s = DC Source Voltage

I = Armature Current

T_e = the electrical torque

k_f = the frictional Constant

J = the rotor inertia

w_m = the Angular Velocity

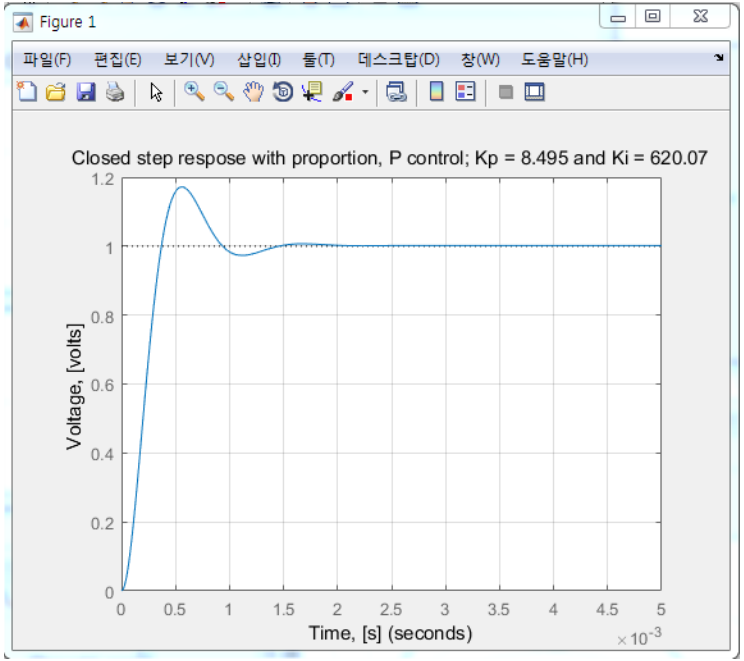
T_L = the Supposed mechanical load

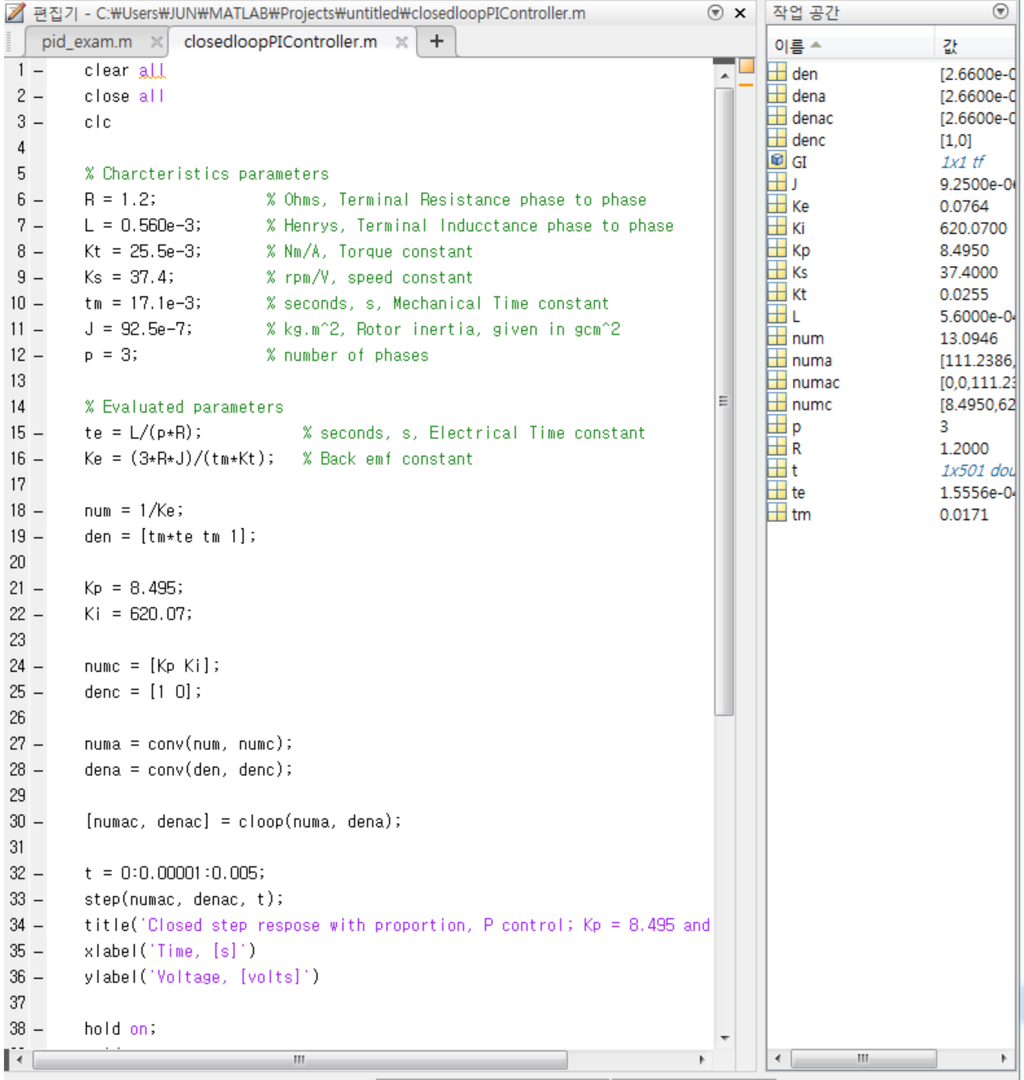
k_e = the back emf Constant

k_t = the torque Constant

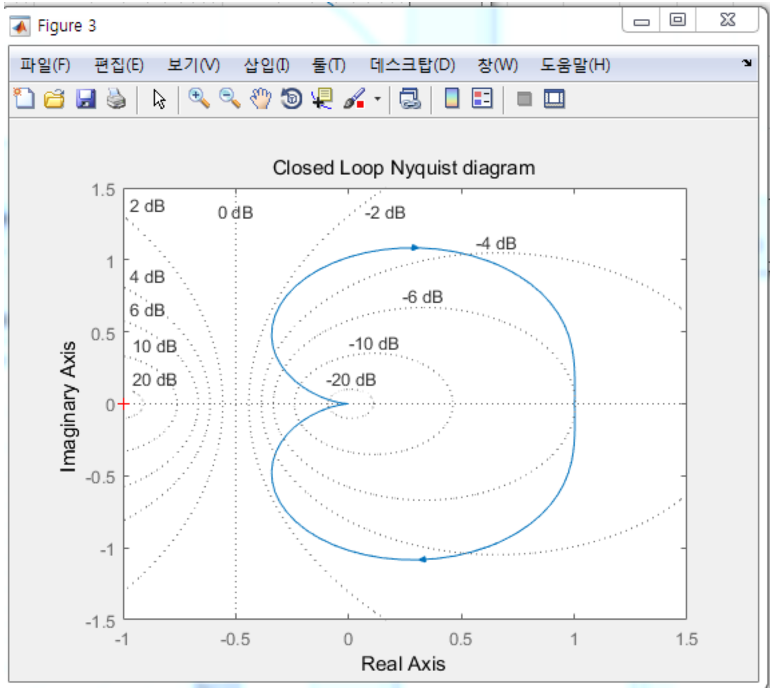
τ_e = the electrical (time Constant)

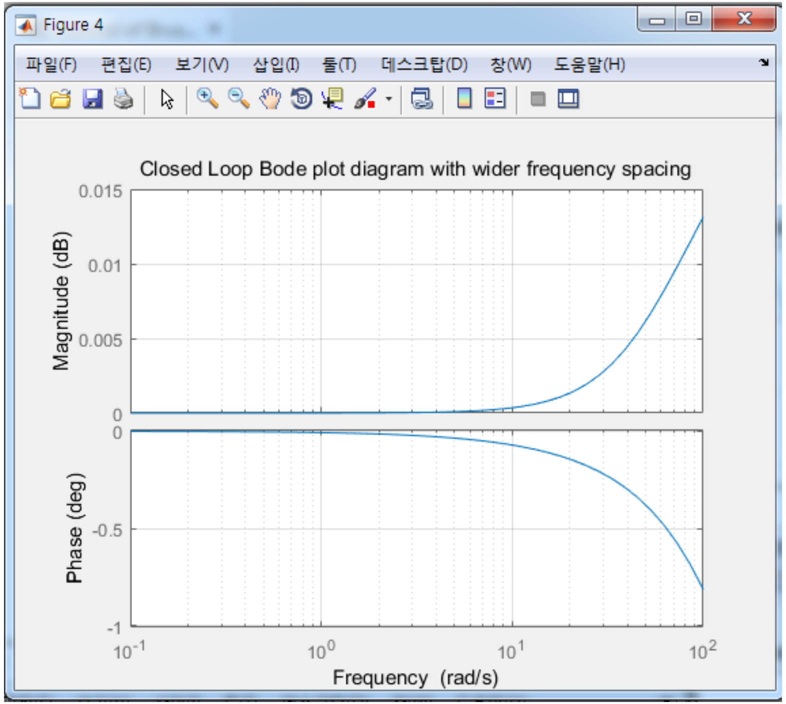
PI Controller (Matlab)





Nyquist, Bode diagram





문제 점

1. BLDC 데이터가 부족하여 plant 값을 구하는데에 가정수치가 너무 많음

- 좀더 찾아보고, 계산할 방법을 찾아본다.

2. 물로켓에 노즐을 구하던지 만들어야 한

Specs:

Magnets: neodymium

Watts: 1800

Poles: 4

Motor Shaft: 5mm with flat spot

Can Measures: 36mm diameter x 60mm long

Weight: 245 Grams

Mounting screw diameter required: 3mm

Mounting Holes 25mm apart for use with most 540 mounts.

할일

-물로켓 부품 구매
-PI controller ccs 구현
-BLDC+ESC+ENCODER+ESC배터리 구동 및 튜닝 마무리
-서보모터용 PID cotroller
-Ship Stabillity 공부(데이터시트)
-자이로 센서, FFT
-SPI 통신을 이용하여 MPU9250 해보기
-온도 센서 구매
-PWM, ADC 공부하기

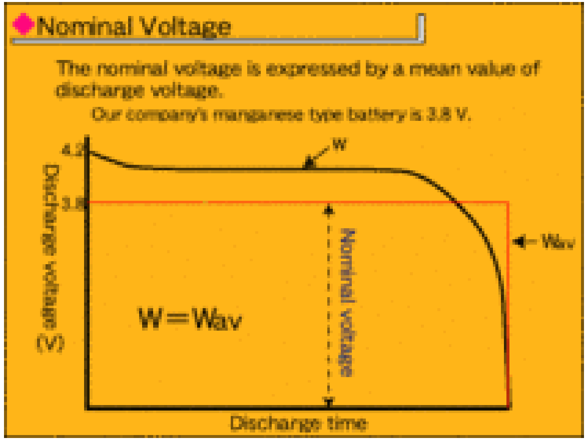
Nominal(공칭) vs Rated(정격)

Rated Voltage : 최대 전압(Maximum Voltage)

Nominal voltage : 평균 전압

(ex, Nominal voltage가 220volts 라면 actual voltage는 5 ~ 10% 높거나 낮다)

둘은 동의어가 될수도 있다. 얼마만큼의 전압을 주느냐에 따라 Nominal Voltage는 바뀐다.



A simple way to look at it is to compare a family car and a sports car, having maximum (rated) speeds of repectively 140km/h and 250 km/h :
If you consider that both cars have to be used on a highway at 120 km/h (nominal speed), the 'rated speed' of the family car will be very close to its 'nominal speed', while there is a big difference between the 'rated speed' and the 'nominal speed' of the sportscar.

정격전압 = 공칭전압 * 1.2/1.1 안전계수 여유율을 상정한 것 같습니다..

stall torque : 순간적으로 낼 수 있는 토크 크기

7.Stall(starting) torque(mNm) 기동 Torque

정격전압에서 모터의기동torque,모터에 전압을 인가하여 정지시킬때 발생하는 torque로서

이론치임(실제로는 모터온도에 따라서 변화함)

모터의 토크는 Stall 토크와 Continuous 토크로 나뉘는데, Stall 토크란 순간적으로 낼 수 있는 토크를 말하고, Continuous 토크는 계속 유지할 수 있는 토크를 말합니다.