



**Automatic Control**

# **PID CONTROL APPLICATION - OPEN-SOURCE AUTOPILOT**

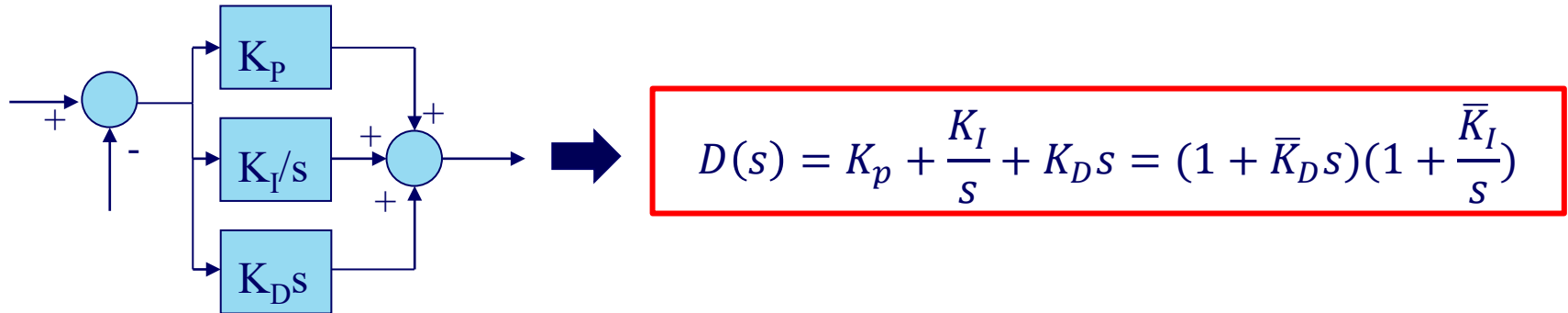




## 9. PID Control

How can we apply?

### ❖ Another formulation of PID Control [1]



- PID control = serial connection of PI control and PD control
  - To design PID control, we can design PI and PD control separately.

### ❖ Another formulation of PID Control [2]

- **+: 비례 이득 결정 후, 적분/미분 이득은 이에 대한 비율로 결정 가능**

$$D(s) = K_p + \frac{K_I}{s} + K_D s = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

where  $T_i = \frac{K_p}{K_I}$  &  $T_d = \frac{K_D}{K_p}$



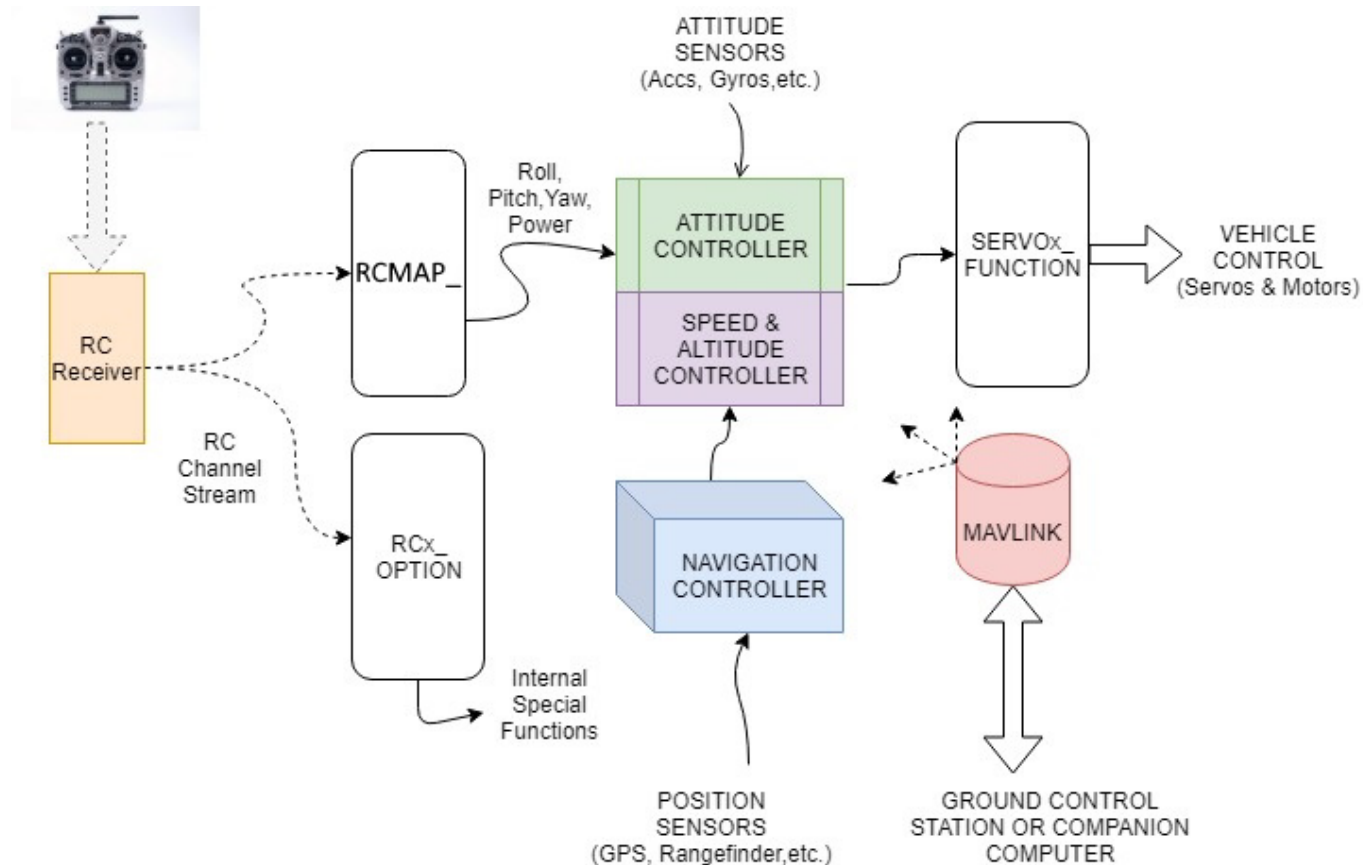
# 9. PID Control

## How can we apply?

### ❖ Example: Open-source autopilot SW “ArduPilot”

- Simple Overview of ArduPilot Operation

- <https://ardupilot.org/copter/docs/common-basic-operation.html>



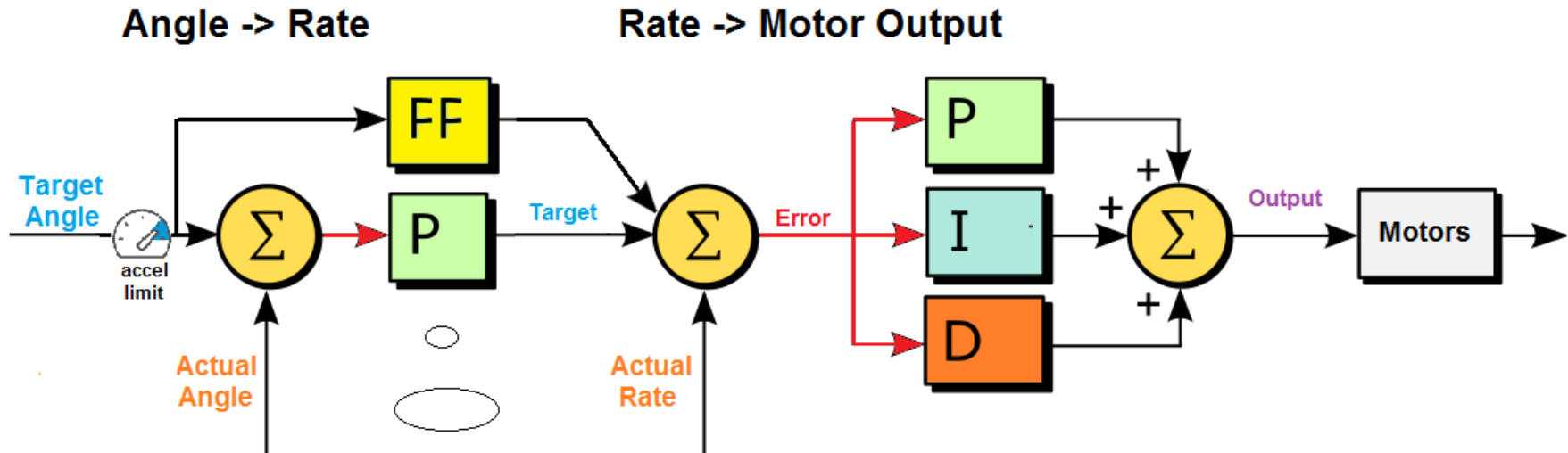


## 9. PID Control

### How can we apply?

#### ❖ Example: Open-source autopilot SW “ArduPilot”

- <https://ardupilot.org/copter/docs/tuning.html>



A block diagram of multicopter attitude control



- <https://ardupilot.org/copter/docs/tuning.html>
- 멀티콥터형 VTOL: 각 축의  $K_p$ 에 대한  $K_p$ 와  $K_I$ 의 상대적 비율 확인!



## 9. PID Control

### How can we apply?

#### ❖ Example: Open-source autopilot SW "Ardupilot"

- <https://ardupilot.org/plane/docs/tuning-quickstart.html>
- 고정익 (Plane): 각 축의  $K_p$ 에 대한  $K_D$ 와  $K_I$ 의 상대적 비율 확인!

The screenshot shows the Ardupilot Configurator interface. The top navigation bar includes tabs for FLIGHT DATA, FLIGHT PLAN, INITIAL SETUP, CONFIG/TUNING (selected), SIMULATION, TERMINAL, HELP, and DONATE. The left sidebar lists options: Flight Modes, APM: Plane Pids (selected), Standard Params, Advanced Params, Full Parameter List, Planner, and Help. The main area displays the following parameters:

Servo Roll Pid	
P	0.160
I	0.000
D	0.020
INT_MAX	15.0

Servo Pitch Pid	
P	0.400
I	0.000
D	0.020
INT_MAX	15.0

Servo Yaw	
Yaw 2 roll	1.0
Integral	0.000
Dampening	0.000
Intergrator Max	15.0

L1 Control - Turn Control	
Period	29.0
Damping	0.800

TECS	
Climb Max (m/s)	5.0
Sink Min (m/s)	2.0
Sink Max (m/s)	5.0
Pitch Dampening	0.000
Time Const	5.0

Other Mix's	
P to T	0
Rudder Mix	0.500

Throttle 0-100%	
Cruise	45.0
Min	30.0
Max	80.0
SlewRate	100.0

Navigation Angles	
Bank Max	35.0
Pitch Max	15.0
Pitch Min	-15.0

Airspeed m/s	
Cruise	18.0
FBW min	15.0
FBW max	25.0
Ratio	3.430

At the bottom, there are buttons for 'Write Params' and 'Refresh Params'.



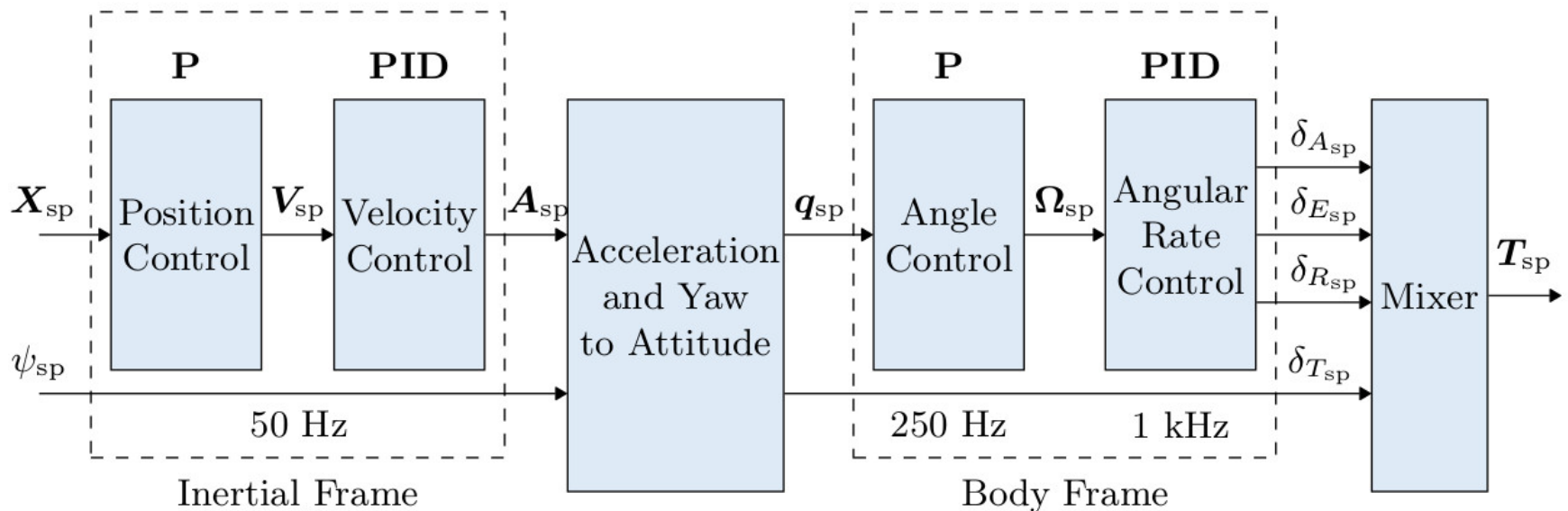
## 9. PID Control

### How can we apply?

#### ❖ Example: Open-source autopilot SW "PX4"

- Multicopter Control Architecture

- [https://docs.px4.io/main/en/flight\\_stack/controller\\_diagrams.html](https://docs.px4.io/main/en/flight_stack/controller_diagrams.html)





# 9. PID Control

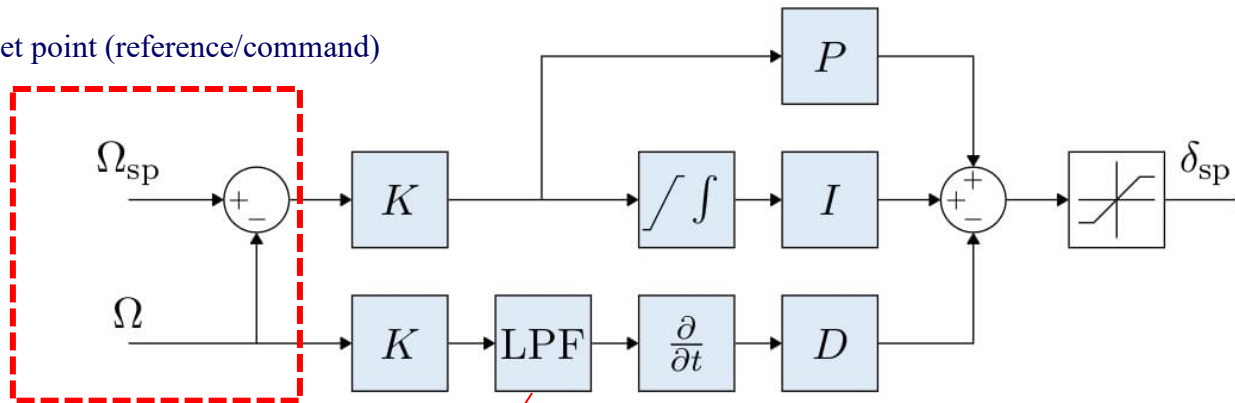
## How can we apply?

### ❖ Example: Open-source autopilot SW "PX4"

#### ■ Multicopter Angular Rate Controller

■ [https://docs.px4.io/main/en/flight\\_stack/controller\\_diagrams.html](https://docs.px4.io/main/en/flight_stack/controller_diagrams.html)

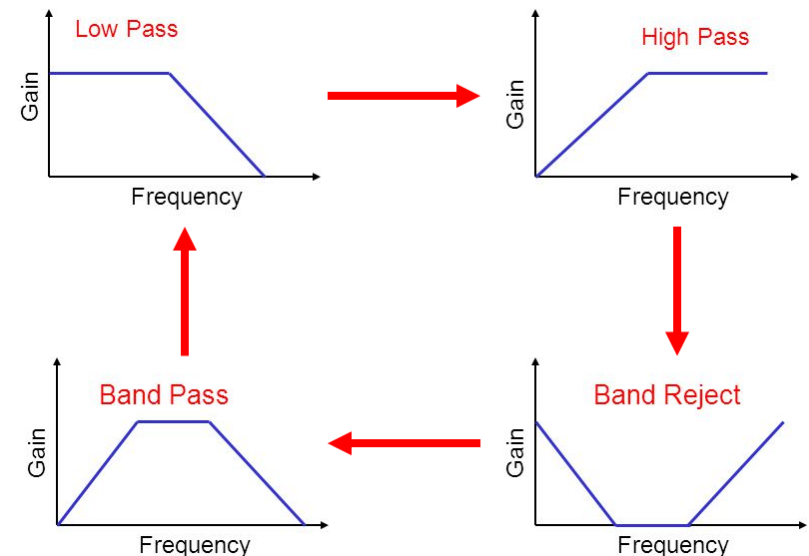
\*sp: set point (reference/command)



Why doesn't  
D-control use  
error?

→ Next page

Why here?  
(Low Pass Filter)





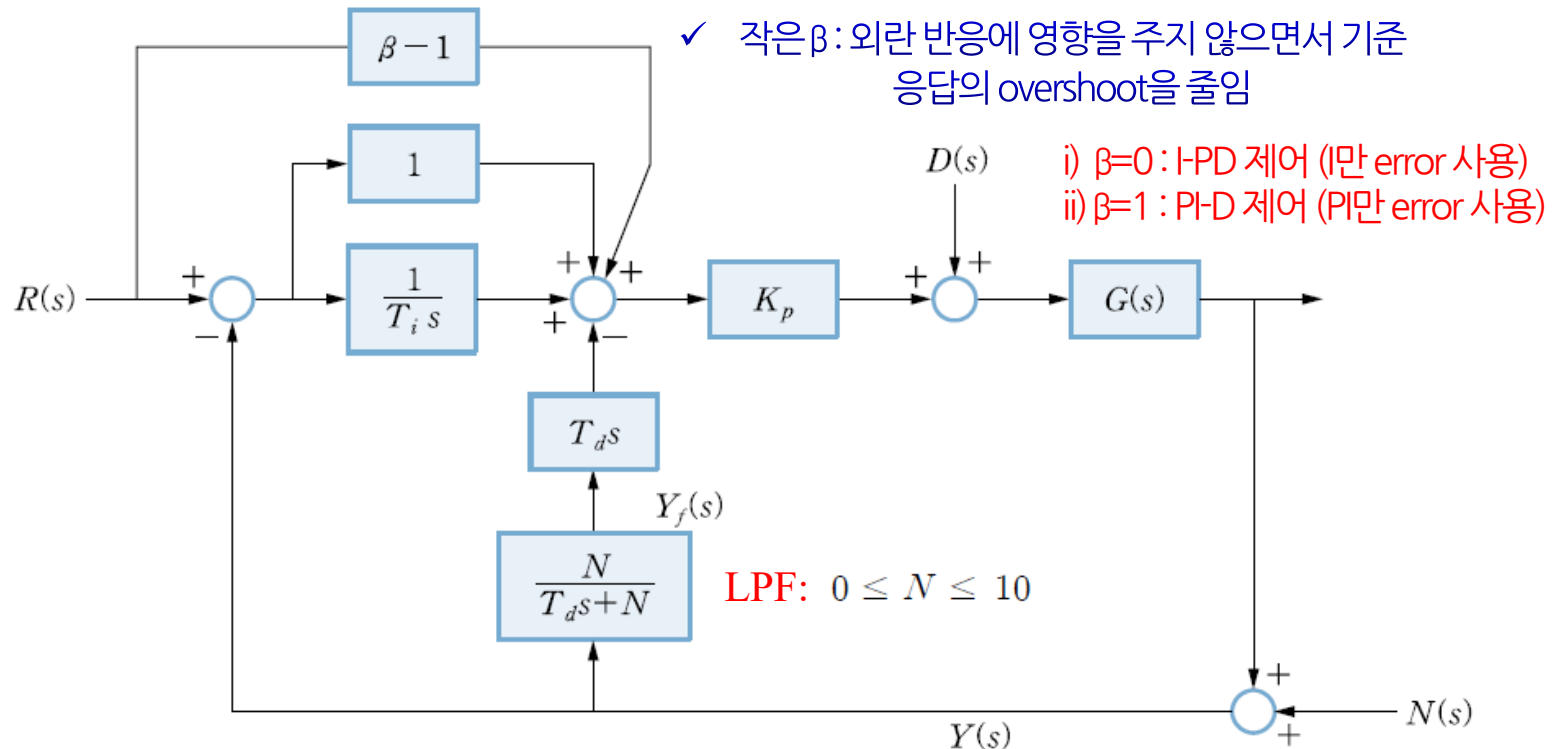


## 9. PID Control

### How can we apply?

#### ❖ Example: Open-source autopilot SW "PX4"

- Multicopter Angular Rate Controller
- **PI-D control: 피드백 신호에만 미분 동작 수행 (PID 제어기 변형)**
  - 기준 신호에 대해 미분 X → 미분 폭주 회피



[그림 6-45] PI-D 제어

$$u(t) = K_p \left[ \beta r(t) - y(t) + \frac{1}{T_i} \int e(t) dt - T_d \frac{d}{dt} y_f(t) \right]$$



# 9. PID Control

## How can we apply?

### ❖ Example: Open-source autopilot SW "PX4"

- [https://docs.px4.io/main/ko/config\\_mc/pid\\_tuning\\_guide\\_multicopter\\_basic.html](https://docs.px4.io/main/ko/config_mc/pid_tuning_guide_multicopter_basic.html)
- 멀티콥터형 VTOL:  $K_p$ 에 대한  $K_D$ 와  $K_I$ 의 상대적 비율/최대치 확인!

Back < ⚙ Vehicle Setup

Summary  
Firmware  
Joystick  
Airframe  
Sensors  
Radio  
Flight Modes  
Power  
Motors  
Safety  
PID Tuning  
Flight Behavior  
Camera  
Parameters

### PID Tuning Setup

Tuning Setup is used to tune the flight controllers.

Rate Controller | Attitude Controller | Velocity Controller | Position Controller

Airmode (disable during tuning) ? Disabled ▾

Thrust curve ? 0.000

Roll Rate

deg/s

sec

■ Response ■ Setpoint

Select Tuning: ☒ Roll ☐ Pitch ☐ Yaw

Overall Multiplier (MC\_ROLLRATE\_K)

0.3 1.1 3

Multiplier for P, I and D gains: increase for more responsiveness, reduce if the rates overshoot (and increasing D does not help).

Differential Gain (MC\_ROLLRATE\_D)

0.0004 0.0032 0.01

Damping: increase to reduce overshoots and oscillations, but not higher than really needed.

Integral Gain (MC\_ROLLRATE\_I)

0.1 0.2 0.5

Generally does not need much adjustment, reduce this when seeing slow oscillations.

Clipboard Values:  
MC\_ROLLRATE\_K 1.1000  
MC\_ROLLRATE\_D 0.0032  
MC\_ROLLRATE\_I 0.200

Save To Clipboard Restore From Clipboard

Clear Start

☐ Automatic Flight Mode Switching



## 9. PID Control

### How can we apply?

#### ❖ Example: Open-source autopilot SW “PX4”

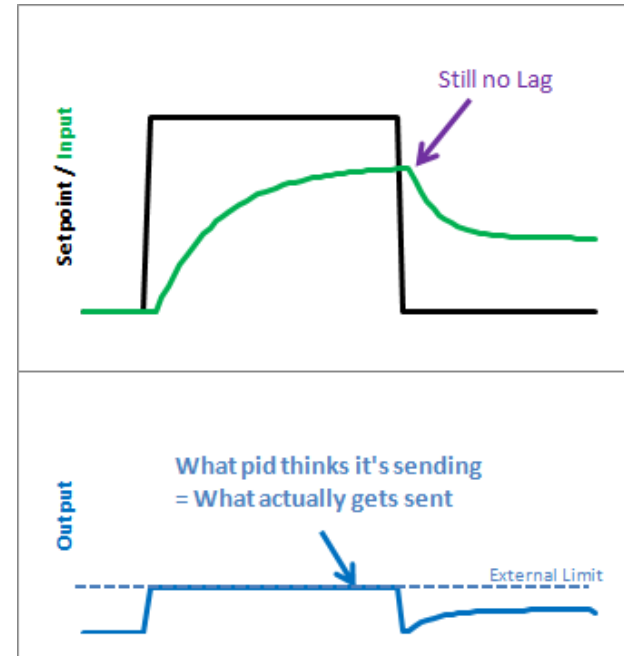
- [https://docs.px4.io/main/ko/config\\_mc/pid\\_tuning\\_guide\\_multicopter\\_basic.html](https://docs.px4.io/main/ko/config_mc/pid_tuning_guide_multicopter_basic.html)

The PID values can be adjusted as follows:

- P (proportional) or K gain:
  - increase this for more responsiveness
  - reduce if the response is overshooting and/or oscillating (up to a certain point increasing the D gain also helps).
- D (derivative) gain:
  - this can be increased to dampen overshoots and oscillations
  - increase this only as much as needed, as it amplifies noise (and can lead to hot motors)
- I (integral) gain:
  - used to reduce steady-state error
  - if too low, the response might never reach the setpoint (e.g. in wind)
  - if too high, slow oscillations can occur



<https://images.app.goo.gl/RHjKcJDWGgi4hKBp7>



Automatic Control

# ANTI-WINDUP CONTROL (적분 누적 방지법)





## 9. PID Control

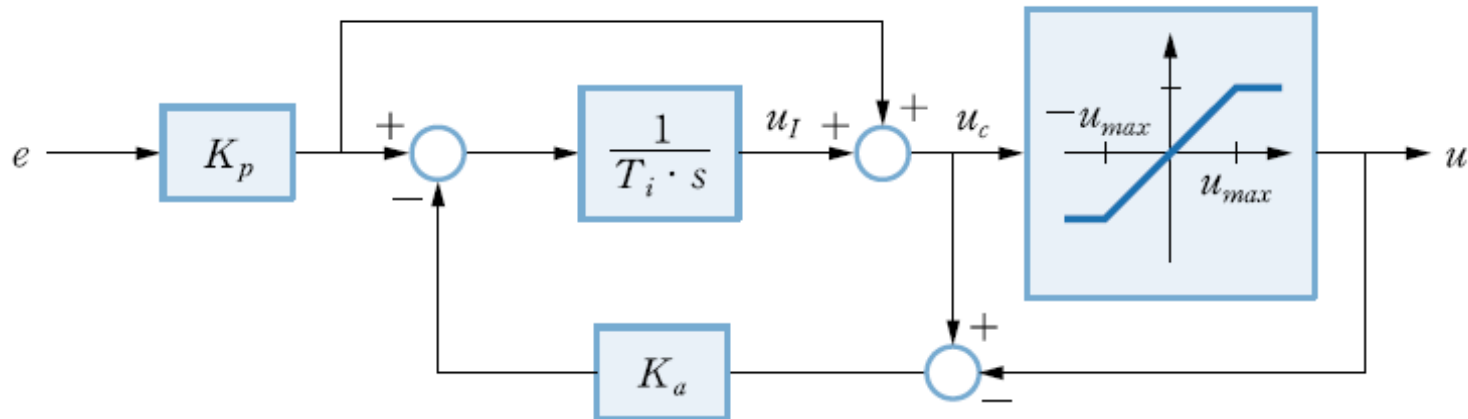
### Integrator Anti-windup (적분 누적 방지)

#### ❖ 적분제어의 문제점

- 대부분의 시스템에서 구동기의 입력값에 한계 존재
- 계속되는 적분 과정으로 한계값 이상의 입력이 요구
  - 입력 포화 → 적분 제어 중지 → 과도응답이 느려지는 현상 발생

#### ❖ 적분 누적 방지 원리

- 제어 신호와 구동기 신호의 차이를 피드백 하여 제어 → 적분 누적 방지



[그림 6-41] 적분 누적 방지를 위한 제어기

$$u_c = -\frac{K_a}{s T_i} (u_c - u) + \left(1 + \frac{1}{s T_i}\right) K_p e$$

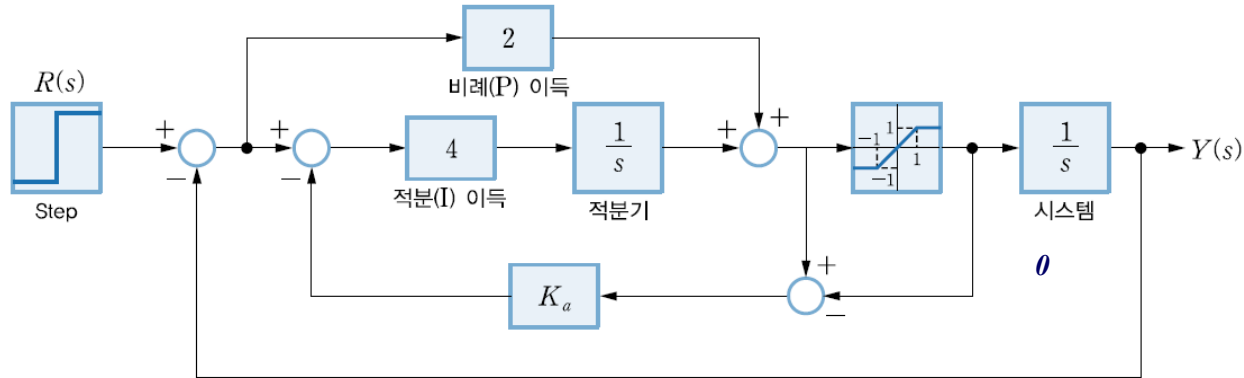


# 9. PID Control

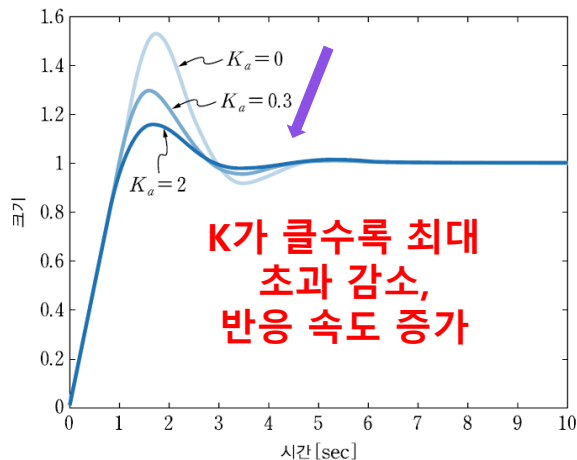
## Integrator Anti-windup (적분 누적 방지)

### ❖ 예제

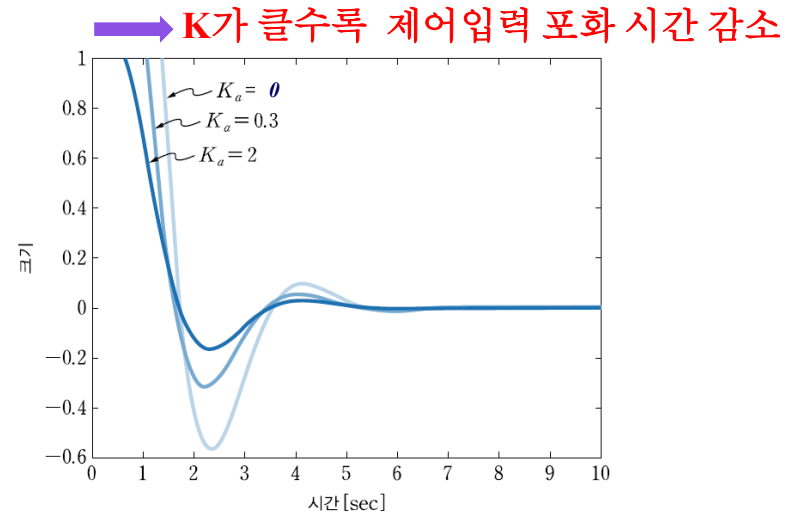
- 적분기 시스템에 대하여 anti-windup을 위한 피드백을 설계하시오.



[그림 6-42] 적분 누적 방지를 위한 적분기



[그림 6-43] 계단 응답에 나타난 적분 누적 방지 효과



[그림 6-44] 제어 입력에 나타난 적분 누적 방지 효과



## 9. PID Control

### Integrator Anti-windup (적분 누적 방지)

#### ❖ [IFAC] Newsletter - April 2024

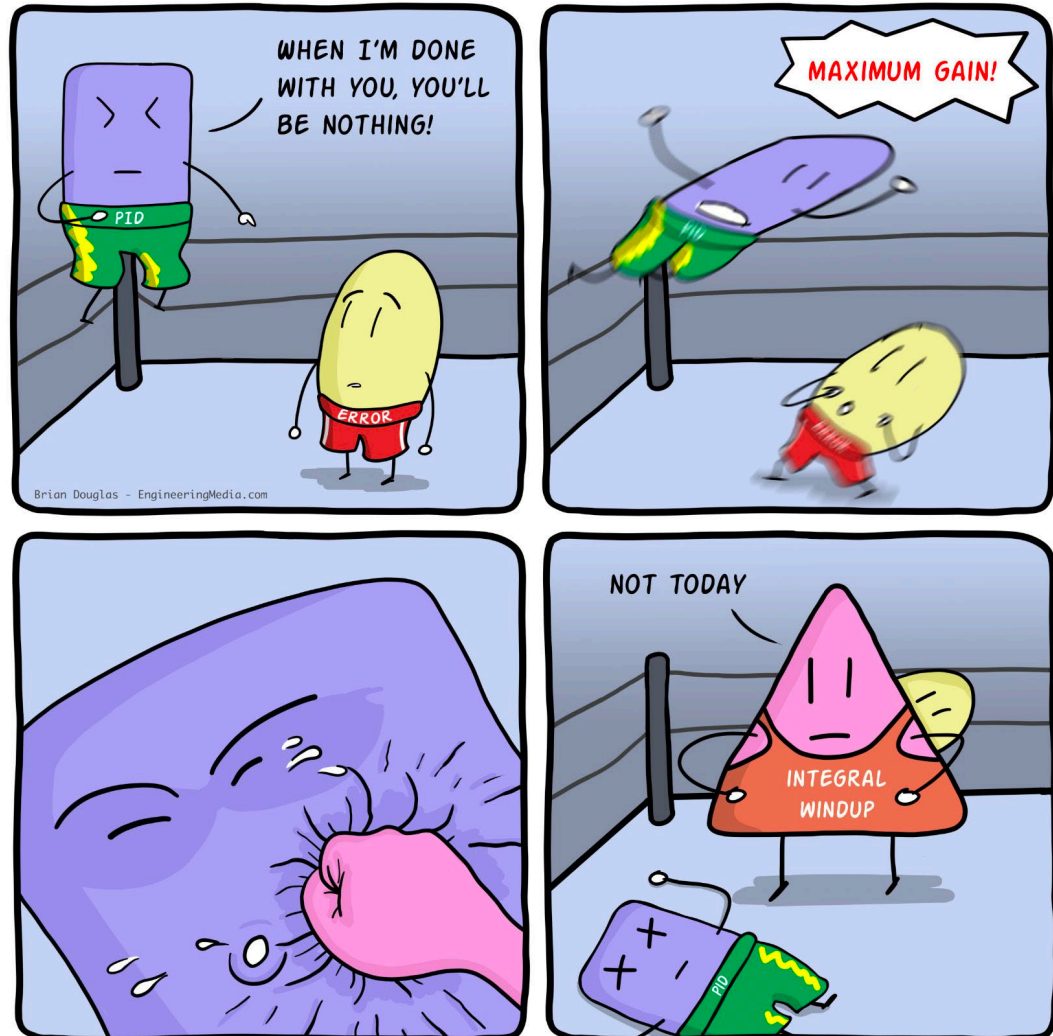
##### Fabulous Control Cartoon

We are pleased to share the second control cartoon in 2024.

It reminds us **not to underestimate windup** in PID control. ;-)

Many thanks to Brian Douglas for preparing the cartoon and for supporting its distribution by IFAC.

<https://www.ifac-control.org/publications/cartoons>



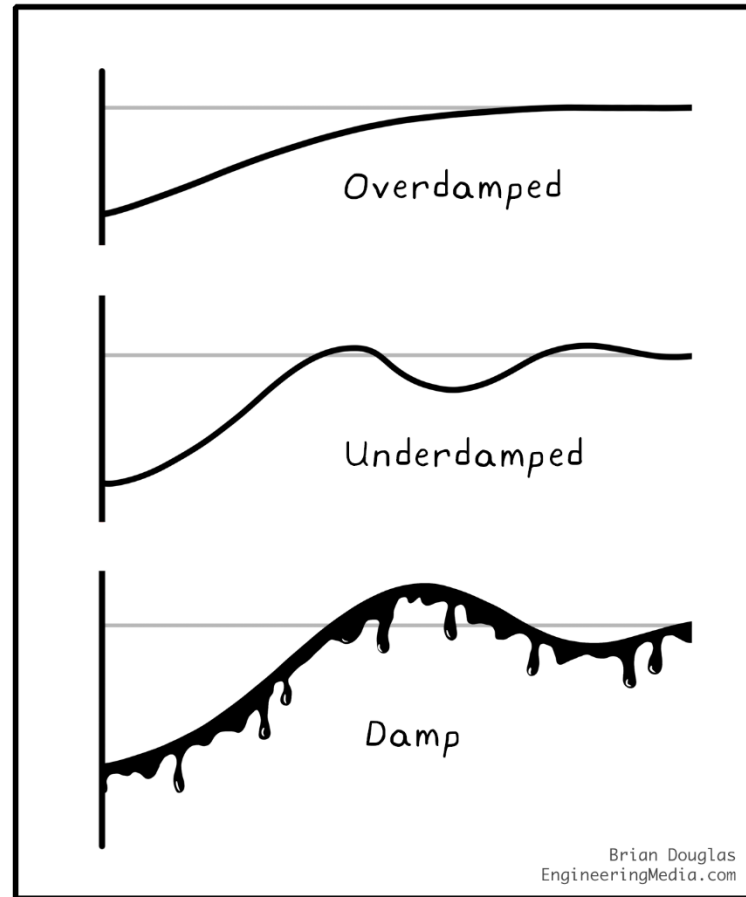


## 9. PID Control

### Integrator Anti-windup (적분 누적 방지)

#### ❖ Another my favorite cartoon

2nd Order Response Curves



It turns out settling time depends on the dynamics of the system and ink viscosity





# Summary

## 9. PID Control

### ❖ PID control application - open-source autopilot

- Another formulation of PID control
- Intro to PID gain setup & tuning for open-source autopilot SW (Ardupilot/PX4)

### ❖ Integrator Anti-windup

- 구동기는 항상 제한 범위가 있다!
  - → 구동기 제한 시 적분기에 대한 입력 줄여주자!



# Thank You !

Automatic Control

