

소프트웨어 프로젝트 2

PID 제어

2024년 2학기

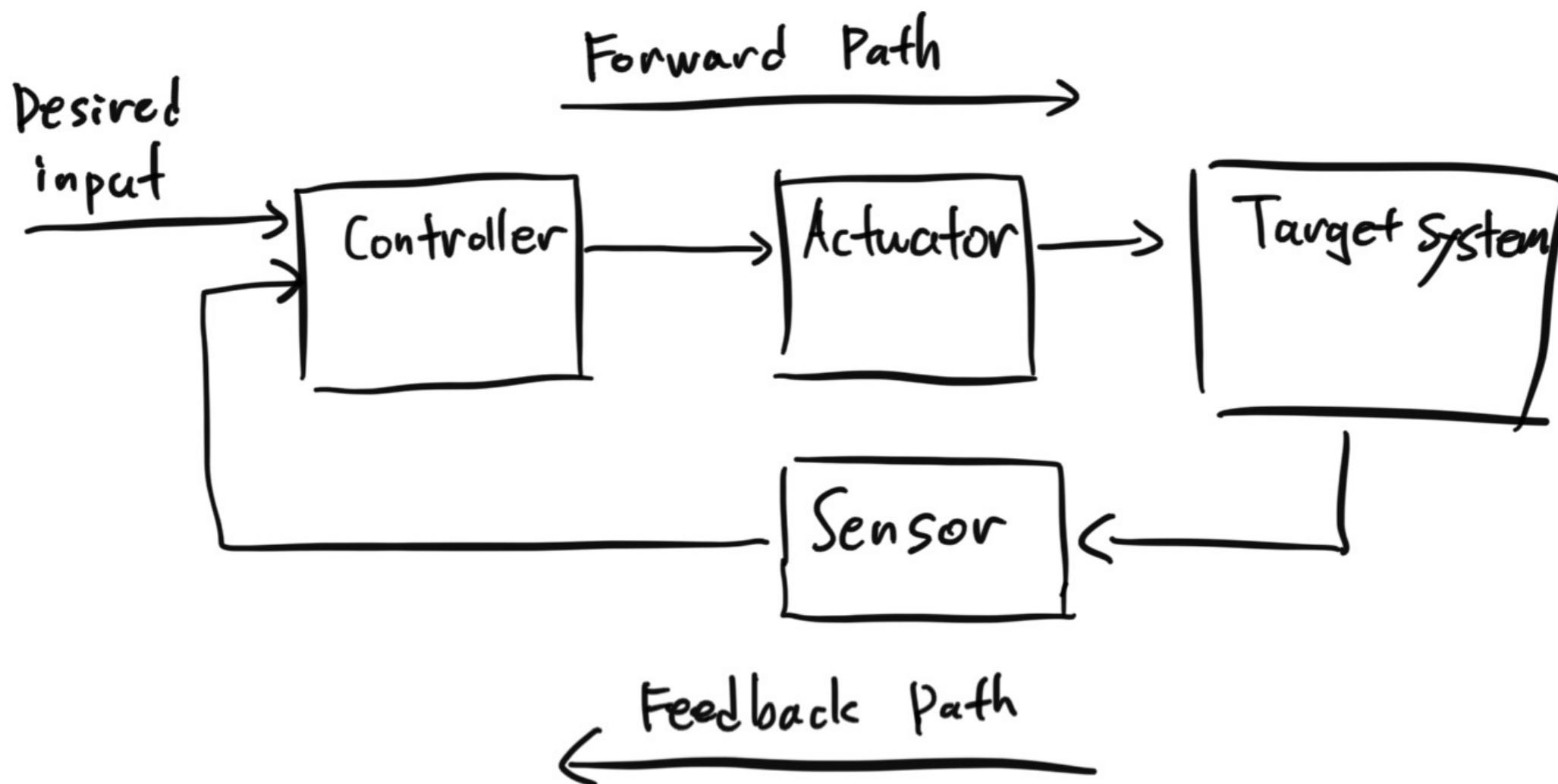
국민대학교

소프트웨어학부/인공지능학부
주용수, 최진우, 한재섭, 허대영

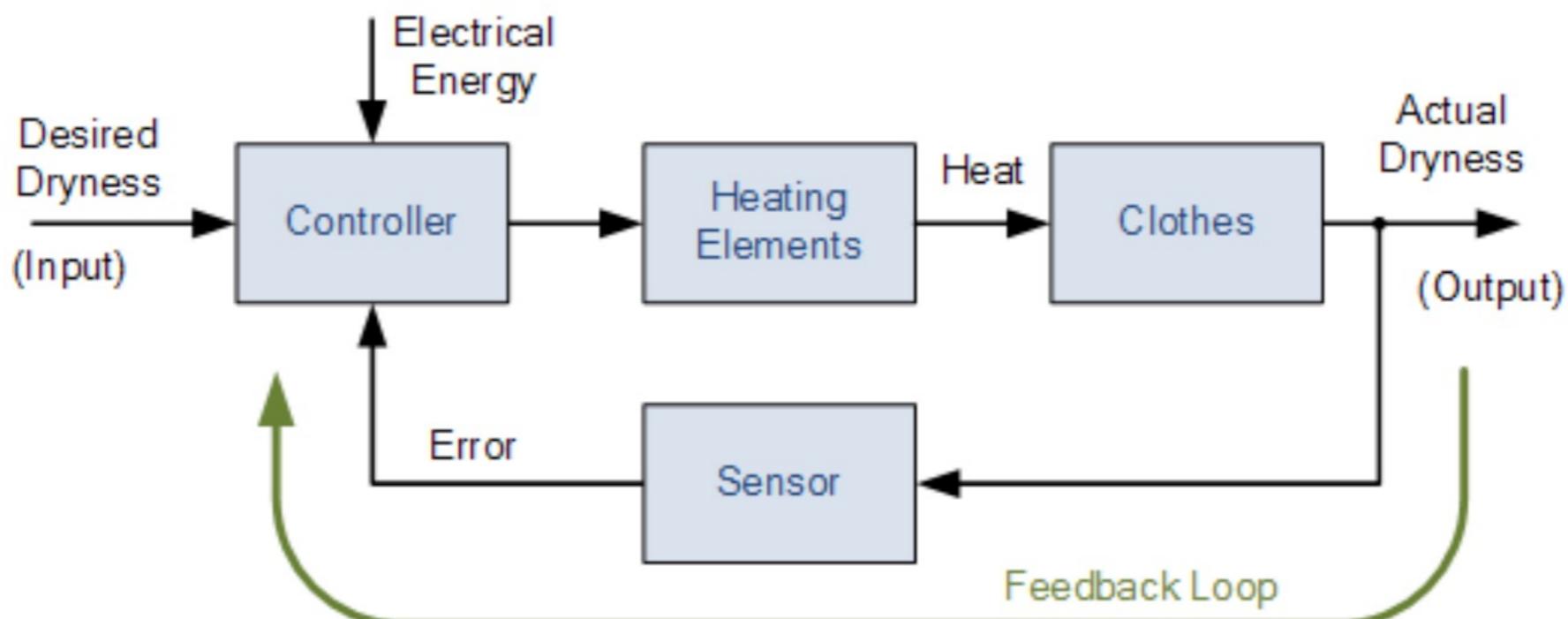
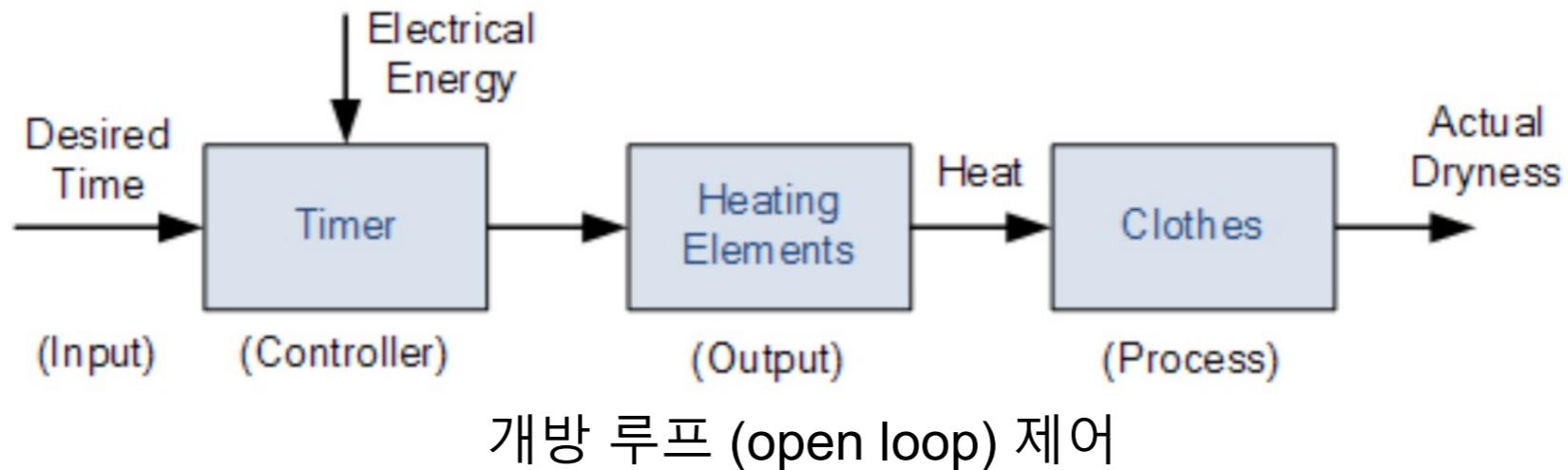
{ysjoo, jaeseob, jnwochoi, dyheo}@kookmin.ac.kr

폐쇄 루프 제어 시스템

- 폐쇄 루프 (closed loop) 제어 시스템
 - A.K.A. 피드백 제어 시스템
 - 시스템의 출력이 입력으로 되먹임(feedback)되는 신호 경로를 사용

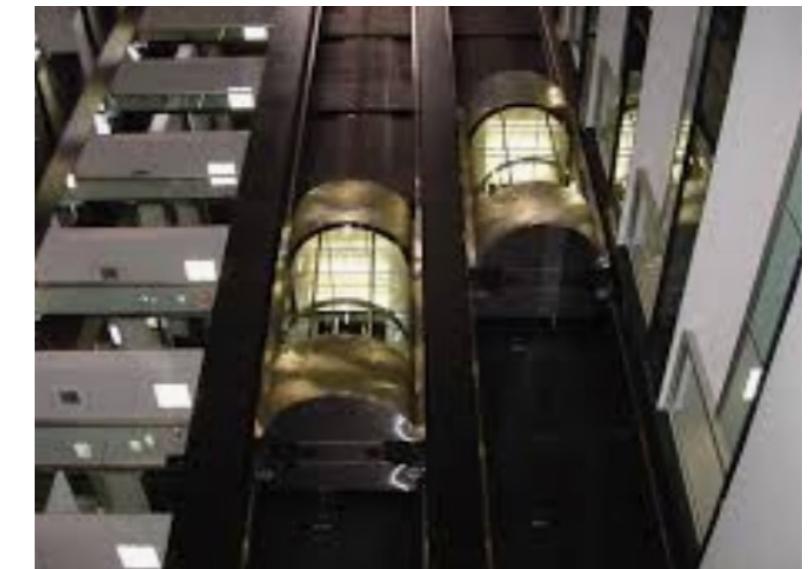


개방 루프 vs. 폐쇄 루프 (의류건조기)



폐쇄 루프 (closed loop) 제어

폐쇄 루프 제어 시스템의 예



폐쇄 루프 제어 시스템의 예

- <https://youtu.be/lmhuvIehB5I>
- <https://youtu.be/UBFBlh3NcBk>
- <https://youtu.be/meMWfva-Jio>
- <https://youtu.be/ItzOya7qWmk>

비례-적분-미분(PID) 제어

- PID 제어
 - 폐쇄 루프 제어 시스템의 일종
 - 목표: 민첩하지 않은(sluggish) 시스템을 원하는 상태로 최대한 신속, 정확하게 제어
- 민첩한 시스템?
 - 무부하 상태의 서보
- 민첩하지 않은 시스템?
 - 서보에 연결된 레일 위의 탁구공

비례-적분-미분(PID) 제어

- PID 제어 원리
 - 비례 이득(proportional gain): 현재 상태에 반응
 - 적분 이득(integral gain): 과거 상태의 누적에 반응
 - 미분 이득(derivative gain): 미래 상태를 예측하여 반응
 - 미래 예측: 최근의 상태 변화 추이를 참조

$$u(t) = K \left(\underbrace{e(t)}_{\text{P}} + \underbrace{\frac{1}{T_i} \int_0^t e(t) dt}_{\text{I}} + \underbrace{T_d \frac{de(t)}{dt}}_{\text{D}} \right)$$

present past future

<https://youtu.be/4Y7zG48uHRo>

<https://www.youtube.com/watch?v=EbT0KJoSfHE>

<https://www.youtube.com/watch?v=KJkqNujb6h4>

PID 제어식의 구성

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right)$$

The amount of control input
Proportional gain
P_term: error
Integral time
I_term: accumulated error
Derivative time
D_term: change of error
Sampling period

Conversion function: weighted state value -> controller strength

PID 구현 예시 1

- 조정 가능 파라미터: K , T_i , T_d
- K 값 변경시 P_{term} , I_{term} , D_{term} 사이의 상대적인 균형을 유지하면서 전체적인 감도를 조절할 수 있음
- I_{term} 의 세기를 줄이려면 T_i 의 값을 키워야 함

```
// Pseudo code
while(1) {
    state_curr = sensor_read();
    error = state_desired - state_curr;
    P_term = error;
    I_term += error * dT;
    D_term = (error - error_prev) / dT;
    output_control = K * ( P_term + I_term / T_i + T_d * D_term);
    error_prev = error;
    sleep(dT);
}
```

PID 구현 예시 2

- 조정 가능 파라미터: **K_p, K_i, K_d**
- 다른 term에 영향을 주지 않고 특정 term의 감도를 조정 가능
- 예시 1, 2 중 선호하는 방식을 선택할 것

```
// Pseudo code
while(1) {
    state_curr = sensor_read();
    error = state_desired - state_curr;
    P_term = error;
    I_term += error * dT;
    D_term = (error - error_prev) / dT;
    output_control = Kp * P_term + Ki * I_term + Kd * D_term;
    error_prev = error;
    sleep(dT);
}
```

PID 제어의 장단점

- 장점
 - Black-box optimization
 - 즉, 목표 시스템 동작에 대한 정확한 이해(accurate system model) 없이도 적용 가능
- 단점
 - 시스템의 최적(가장 최선의) 제어를 보장하지 못함

PID 파라미터 설정

- PID 제어식에는 설정이 필요한 계수가 포함됨
- 최적의 제어 결과를 얻을 수 있는
시스템적 절차(systematic way)를 정의하기 어려움
- 시행 착오(trial & error)
 - 다양한 설정값을 시도하면서 제어 품질을 평가하여 설정값 조정
- 경험적 방법(empirical method)
 - <https://robotics.stackexchange.com/questions/167/what-are-good-strategies-for-tuning-pid-loops>

PID 설정값 조정

- 보다 정교한 휴리스틱
 - Ziegler-Nichols method
 - https://en.wikipedia.org/wiki/Ziegler%E2%80%93Nichols_method
 - Twiddle algorithm
 - <https://www.youtube.com/watch?v=2uQ2BSzDvXs>
 - <https://martin-thoma.com/twiddle/>
- 상용 소프트웨어
 - CEMTool / MATLAB & Simulink

적분기 와인드업(Windup)

- 액추에이터의 조작 범위는 유한함
- 포화상태(Saturation):
제어값이 액추에이터 조작 범위를 넘어가는 경우
 - 와인드업: 포화상태가 지속되는 경우 누적오류 값이 액추에이터가 감당하기 어려운 정도로 커질 수 있음
 - 목표 상태에 도달하였음에도 불구하고 적분항의 누적오류로 인해 액추에이터에 과도한 제어값이 입력됨
- 해결책
 - 와인드업 발생시 적분항을 미리 설정된 값(예: zero)으로 초기화
 - 희망 상태값의 급격한 변화를 지양하여 오류 누적을 방지
 - 적분항을 일시적으로 비활성화
 - 적분항의 최소-최대값 범위를 제한
 - FYI: <https://controlguru.com/table-of-contents/>

적분기 와인드업(Windup)

- 적분항 최대-최소 범위 제한 예

```
1 // Pseudo code
2
3 while(1) {
4     state_curr = sensor_read();
5     error = state_desired - state_curr;
6     P_term = error;
7     I_term += error * dT;
8
9     if (I_term > _ITERM_MAX)
10        I_term = _ITERM_MAX;
11    if( I_term < - _ITERM_MAX)
12        I_term = - _ITERM_MAX;
13
14    D_term = (error - error_prev) / dT;
15    output_control = K * ( P_term + I_term / T_i + T_d * D_term);
16    error_prev = error;
17    sleep(dT);
18 }
```

유용한 링크

- <https://robotics.stackexchange.com/questions/167/what-are-good-strategies-for-tuning-pid-loops>
- <https://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf>
- 한글 블로그 자료
 - <https://m.blog.naver.com/droneaje/221989463375>
 - https://m.blog.naver.com/ann_arbor/221354662048