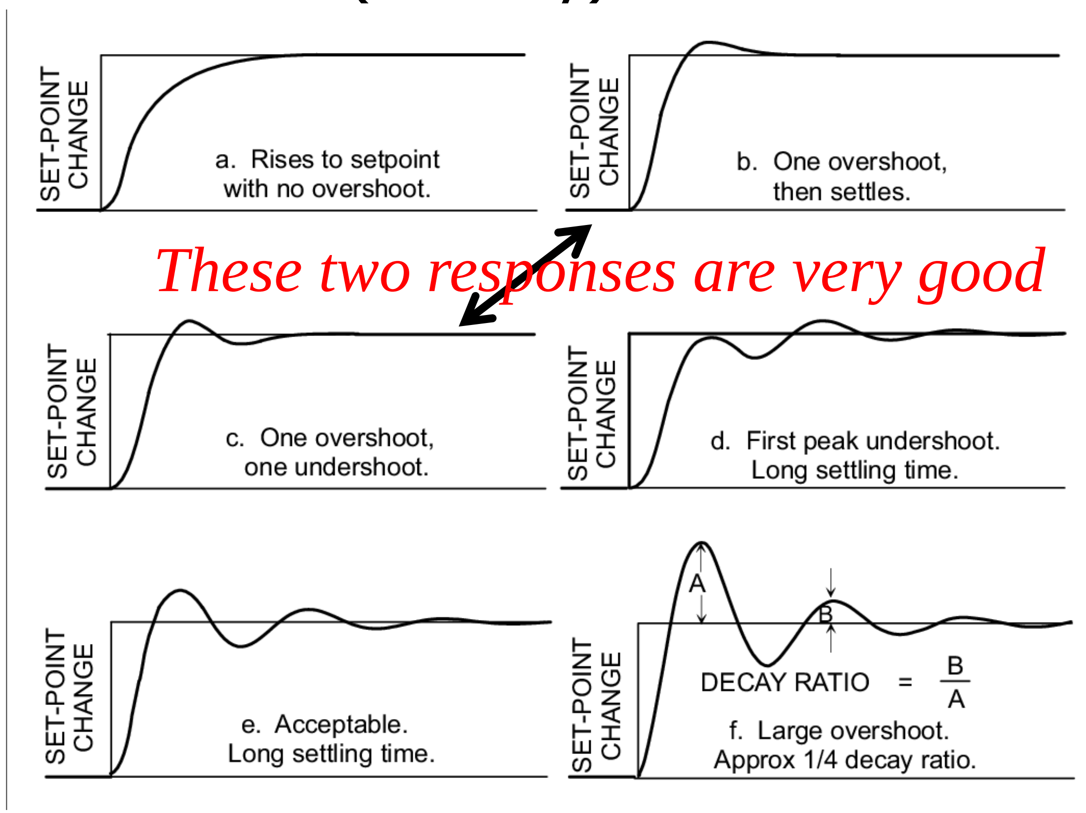
PID Tuning

CSE 425 Industrial Process Control

Lecture 6

Controller Tuning

- Tuning is about adjusting the PID controller parameters $K_c, T_i,$ and $T_d,$ to give "best" response.
- It is a compromise between minimizing settling time (speed) & overshoot (stability).

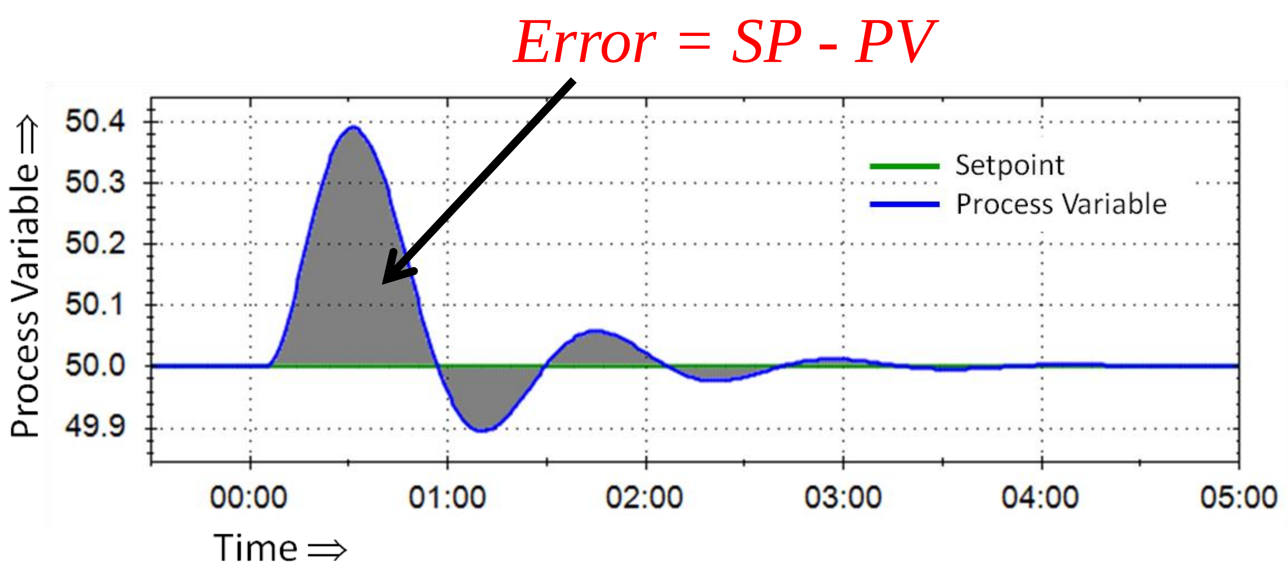


How to judge step response?

- Overshoot: acceptable 5-10%
- Decay ratio = 2^{nd} peak / 1^{st} peak, the smaller the quicker the response
  - If the response makes one or two cycles before reaching setpoint, we make a good job!

Error Integral Criteria

- Error criteria penalize both Size & Duration of Error
  - Integral of Squared Error (ISE)
  - Integral of Absolute Error (IAE)
  - Integral of Time-weighted Absolute Error (ITAE): penalizes late errors (gives best settling time)



$$ISE = \int_0^\infty e^2(t)dt$$

$$IAE = \int_0^\infty |e(t)|\,dt$$

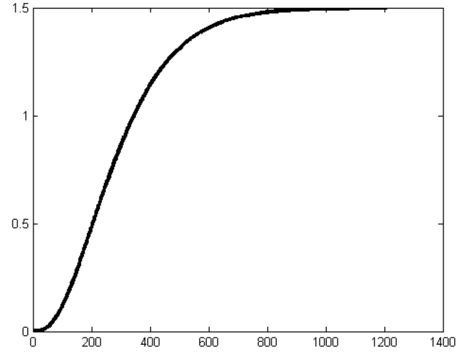
$$ITAE = \int_0^\infty t\,|e(t)|\,dt$$

PID Controller Tuning

- Tuning can be done by trial and error
  - time-consuming
  - costly
- There are many *systematic* methods for PID tuning such as classical Ziegler–Nichols (ZN) tuning rules.
- We will study ZN's two methods.

ZN's first method

1) Start at steady state
2) Turn the controller into Manual mode (***open-loop***)
3) Apply a step input, *of suitable magnitude,* to the process.
4) Record the step response (also called process reaction curve). It has to be *S*-shaped in order for this method to be applicable.
5) Fit a first-order plus dead time (FOPDT) model to the curve.



Two-points method

- The *Two-points* method is used to fit FOPDT model to process reaction curve

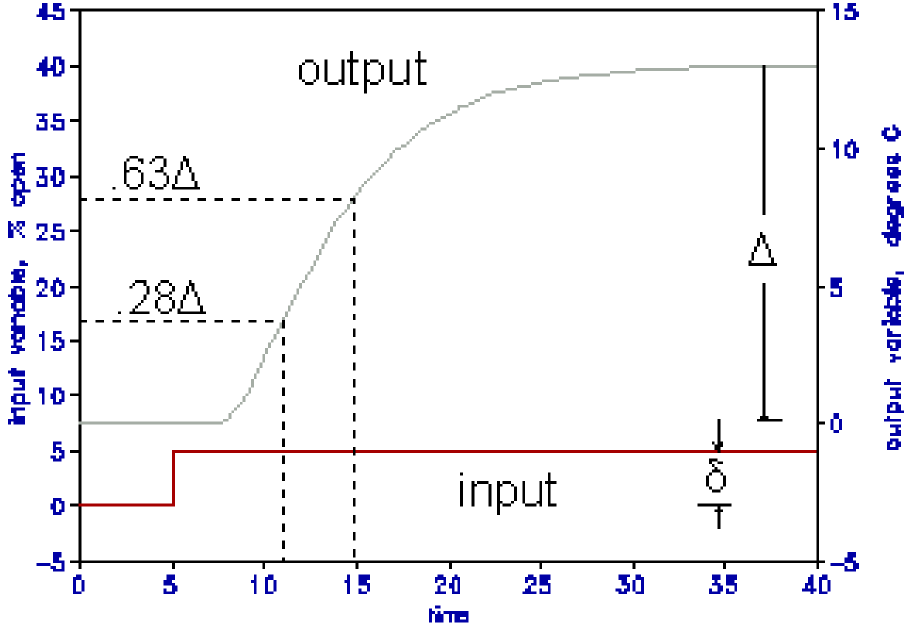
$$G(s) = \frac{K_p e^{-\theta s}}{\tau s + 1},$$

Where θ is time delay, τ is time constant, and K_p is dc gain.

$$K_p = \frac{\Delta}{\delta} = \frac{\textit{total output change}}{\textit{total input change}}$$

$$\tau = 1.5\,(t_{63\%} - t_{28\%})$$

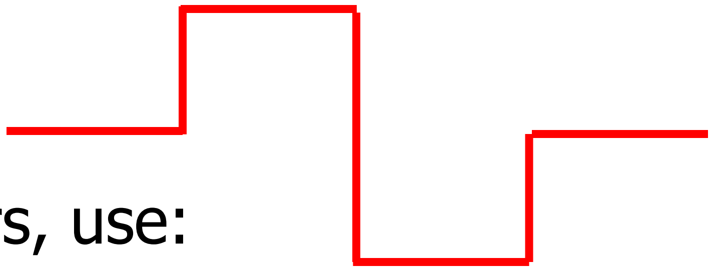
$$\theta = t_{63\%} - \tau$$



- Times t_{28} and t_{63} are measured from the time step input is applied
- Remember K_p is dimensionless

Process Identification

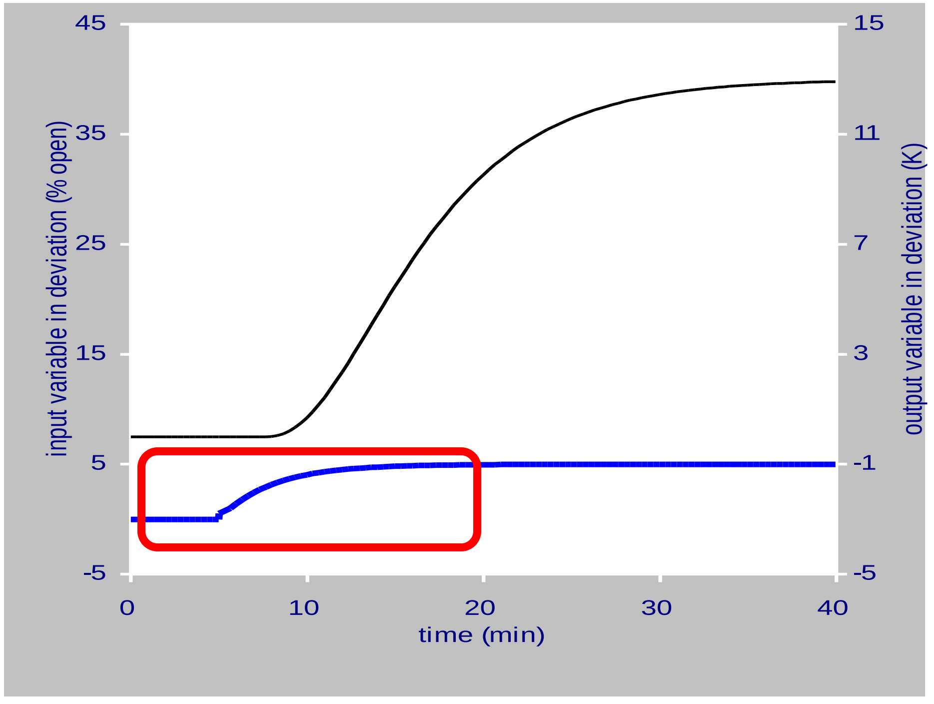
- The step test is also called Bump test: 5% (Avoid 0-100% change)
- Stay around the middle of normal range of operation to avoid nonlinearity of the valve.
- Apply more than one bump test as shown below; these are three steps.



- If the tests give different parameters, use:
  - largest process gain
  - longest dead time
  - shortest time constant
  - ➢ This gives conservative tuning

Bad Bump Test

- In the following figure, the input is not really a step.
- Hence, we can not extract correct information about the time delay and time constant of the process



Dead Time-to-Lag Ratio (θ/τ)

- This ratio tells us how much is the process easy to control
  - $\theta/\tau < 0.33$ easy to control
  - $0.33 < \theta/\tau < 1.0$ moderate
  - $1.0 < \theta/\tau$ difficult to control
- The controller gain is inversely proportional to θ/τ.
  - if θ/τ is large, the gain should be low; i.e. the controller must be cautious

ZN Tuning rules

- After having identified the process parameters, we use the following table to find the P, PI, or PID controller parameters corresponding to the FOPDT model obtained.
- Note how adding *Derivative* action allows increasing controller gains to obtain faster response.

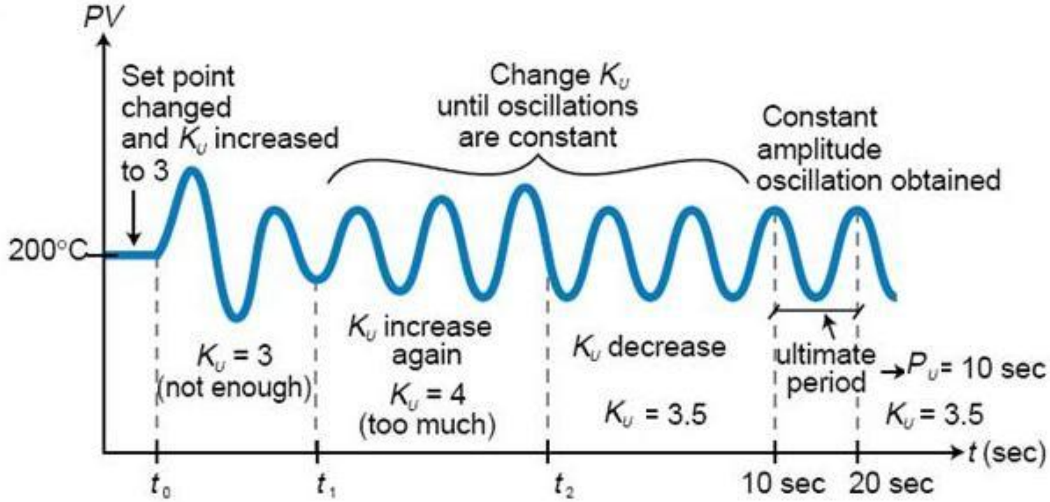
Controller	K_c	T_i	T_d
P	$\frac{1}{K_p(\theta/\tau)}$	∞	0
PI	$\frac{0.9}{K_p(\theta/\tau)}$	3.3θ	0
PID	$\frac{1.2}{K_p(\theta/\tau)}$	2θ	0.5θ

ZN’s Second Method

- If the process reaction curve is not s-shaped or the process is open loop unstable, the first method is not applicable.
- In this case, we resort to Ziegler Nichols’ second method also called the Ultimate-Cycle method.
- In contrast to the first method, the ultimate cycle method is used in closed-loop.

Procedure

1. Put the process under closed-loop proportional control.
2. Create a small disturbance in the loop by changing the set point.
3. Adjust the proportional gain, increasing and/or decreasing, until the response shows oscillations with constant amplitude (a sustained oscillation called *hunting*).
4. Record the gain value (K_u) and period of oscillation (T_u).



ZN's 2nd Tuning rules

Use the following table to find controller parameters.

Controller	K_c	T_i	T_d
P	$0.5K_u$	∞	0
PI	$0.455K_u$	$0.833\,T_u$	0
PID	$0.6K_u$	$0.5\,T_u$	$0.125\,T_u$

Simulation Example

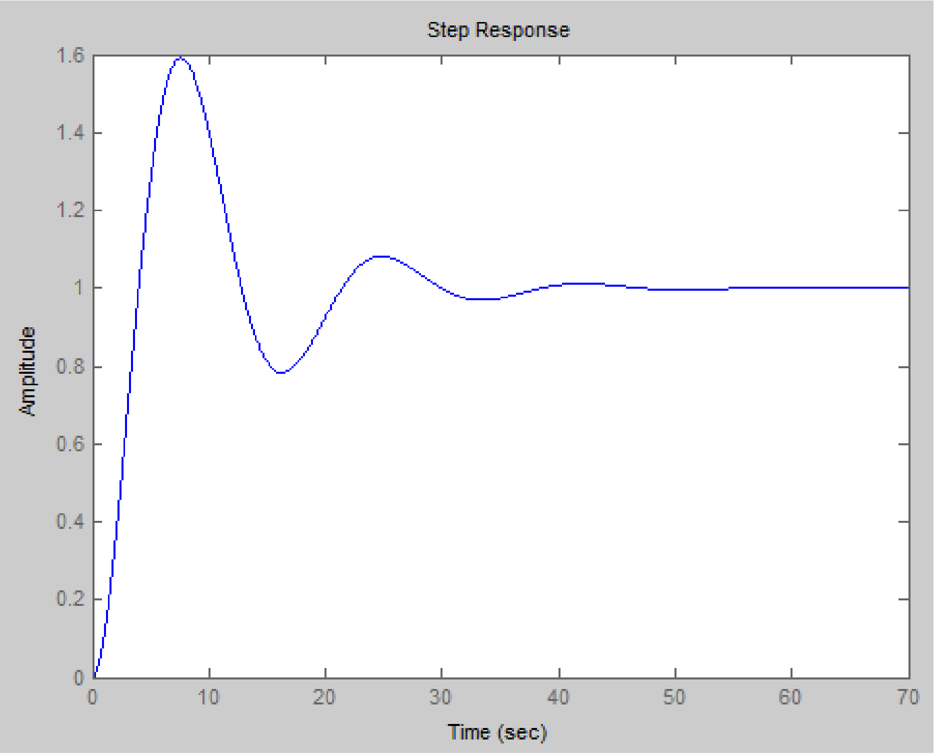
- Consider the following process

$$G(s) = \frac{1}{s(2s+1)^2}$$

- As the process has an integrator, we use ZN's 2nd method.
- Using simulation, we can find that K_u=1 and P_u=12.54.
- Let us simulate the closed-loop set-point step response using ZN-tuned PID controller.

MATLAB Code

```
% MATLAB code
t=0:0.01:70;
s=tf('s');
G = 1/(s*(2*s+1)^2);
% Process parameters
Ku = 1; Pu = 12.54;
 %  PID parameters, Method 2
Kc = 0.6*Ku;    Ti = 0.5*Pu;
Td = 0.125*Pu;
Gc = pid(Kc, Kc/Ti, Kc*Td);
% Set point step response
cloop = Gc*G/(1+Gc*G);
figure(2)
step(cloop,t)
```



Comments

- We see that the response shows large overshoot (~60%) which is not acceptable.
  - In fact, ZN are designed to give quarter-cycle decay ratio.
  - This gives good disturbance rejection but exhibits large overshoot for set point response.
- Despite this drawback, ZN tuning serves as a good starting point for fine tuning.
- Many other methods are available in the literature (e.g. those minimizing ITAE or ISE criteria)

Set point vs. Disturbance Responses

- When a setpoint change occurs, the controller sees quick and big error signal.
- On the contrary, disturbance causes small gradual error

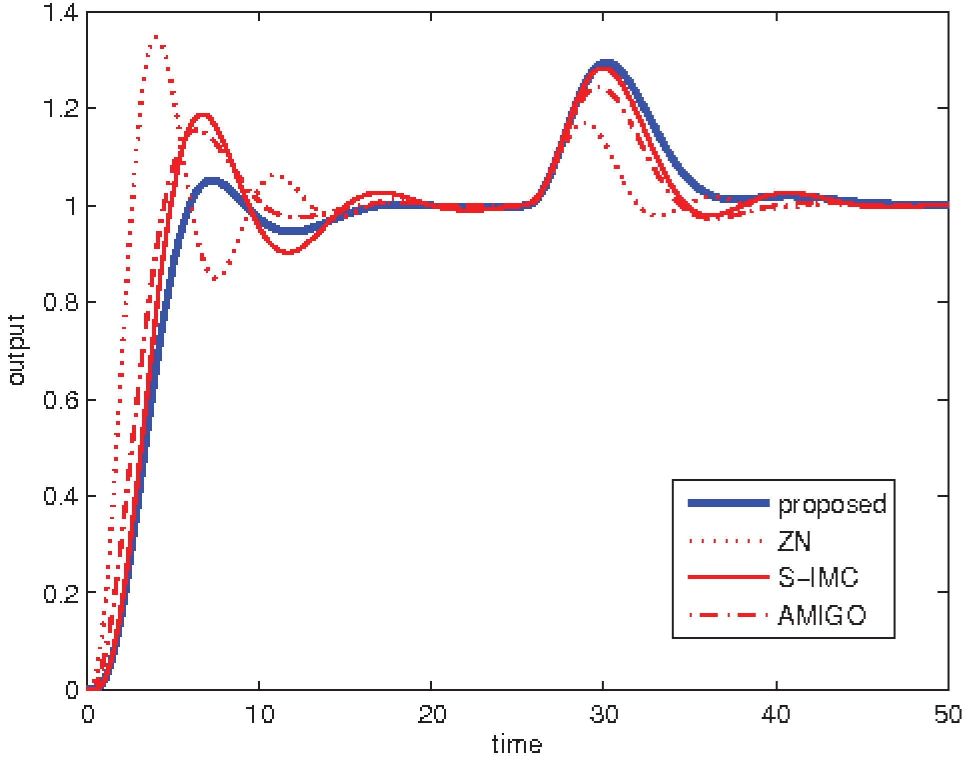
<u>In general:</u>

Tuning for setpoint changes gives a relaxed controller. This causes very slow load rejection.

Tuning for load rejection is aggressive (higher gain and less T_i). This tuning causes large overshoot for setpoint changes.

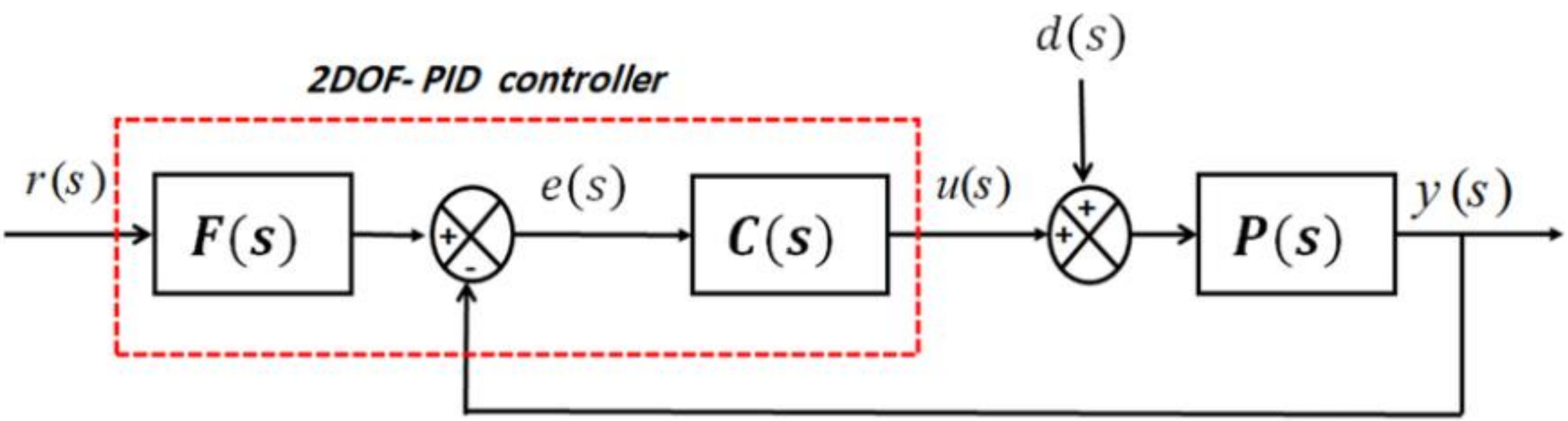
Set point vs. Disturbance Responses

- For example, in the following figures, there are several tuning methods.
- We can see that the best method for setpoint tracking is the worse in disturbance rejection and vice versa.



Setpoint Filtering or Softening

- To achieve good setpoint and disturbance responses simultaneously, we tune the controller for load disturbance rejection (i.e., make the controller aggressive) and apply the setpoint through a filter to avoid large overshoot in setpoint response.
- Note the error is E = SP_f-PV



Setpoint Filtering

- With set point filtering, the controller is said to have Two Degrees of Freedom (2DOF)
- Setpoint filtering has no effect on load response