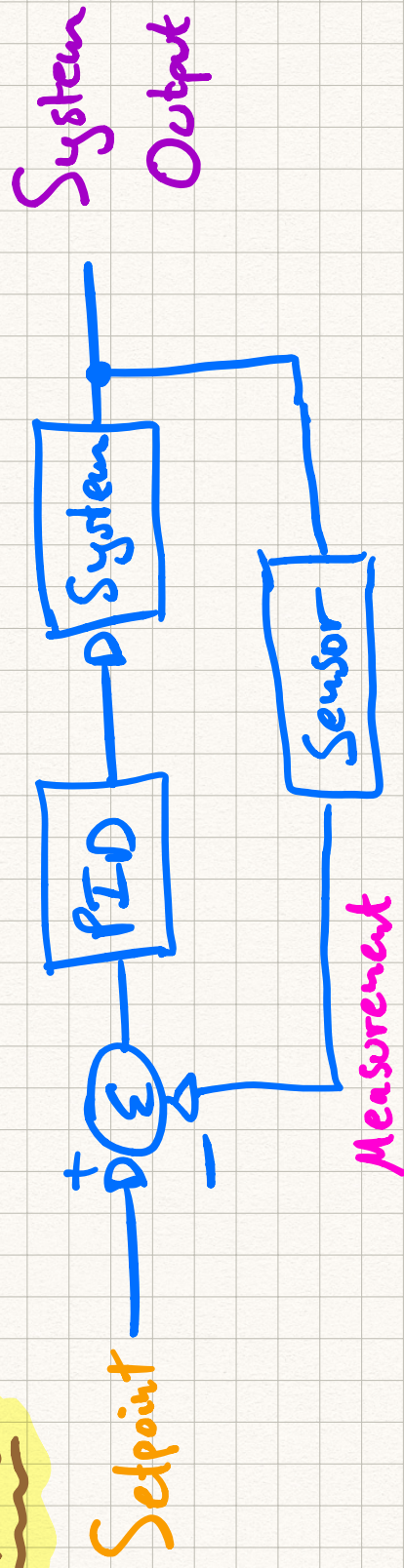


PID Controller Implementation in Software

- Overview.
- Converting from continuous to discrete time.
- Practical considerations.
- PID code in C (for embedded systems).
- Example application.

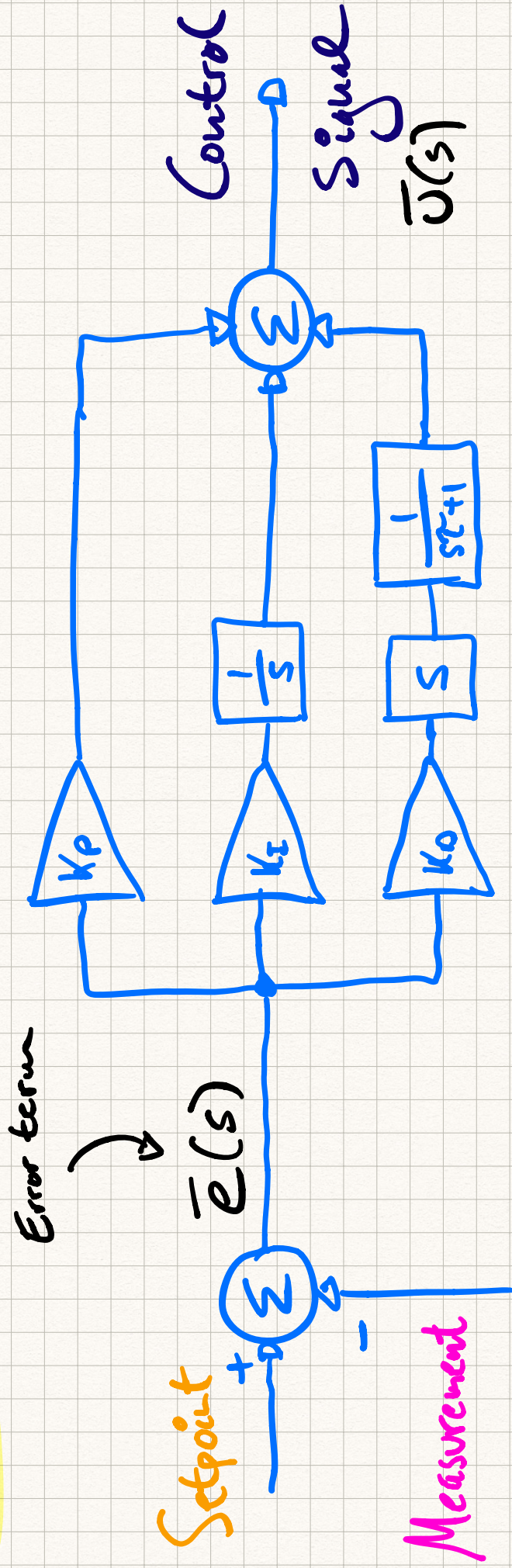
Overview



- Want to make system output track desired setpoint via feedback!
- PID controller takes two inputs and produces a control signal.
 - ↳ setpoint
 - ↳ measurement
- System usually in continuous time, PID controller usually implemented digitally.

⇒ How can we write the controller in code?

PID in Continuous Domain



$$G(s) = \frac{\bar{u}(s)}{\bar{e}(s)} = \underbrace{K_P}_{\text{Proportional}} + \underbrace{K_I \cdot \frac{1}{s}}_{\text{Integral}} + \underbrace{K_D \cdot \frac{s}{s+1}}_{\text{(Filtered) Derivative}}$$

Continuous to Discrete

Can implement this in code!

→ I.e. s-Domain to z-Domain to difference equation!

→ Use Tustin transform (best frequency domain match)

① Substitute $s \rightarrow \frac{2}{T} \frac{z-1}{z+1}$

② Recall: $\bar{Y}(z) = \bar{X}(z) \cdot \bar{z}^{-1} \Rightarrow y[n] = x[n-1]$

($T \hat{=}$ sampling time of discrete controller in seconds)

After a lot of substituting and rearranging...

$$\begin{bmatrix} p \\ i \\ d \end{bmatrix} = \begin{bmatrix} K_p \cdot e[n] \\ \frac{K_I T}{2} (e[n] + e[n-1]) + i[n-1] \\ \frac{2K_D}{2\tau + T} (e[n] - e[n-1]) + \frac{2\tau - T}{2\tau + T} d[n-1] \end{bmatrix}$$

$$\Rightarrow u[n] = p[n] + i[n] + d[n]$$

↖ Controller output

Practical Considerations

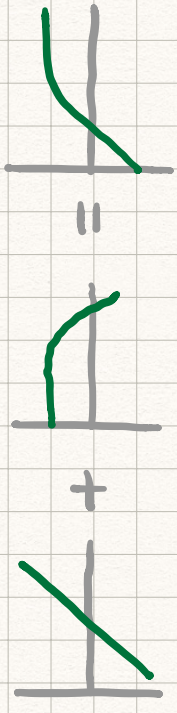
→ Derivative amplifies HF noise.

→ Derivative "kick" during setpoint change.

→ Integrator can saturate output.

→ Limits on system input amplitude.

→ Choosing a sample time T .



Derivative filter!

Derivative-on-measurement!

Integrator anti-windup!



Clamp controller output!

$$T_{\text{controller}} < \frac{T_{\text{BW, system}}}{10}$$

Code Structure

- Header-file "library". (`#include "pid.h"`)
- PID controller struct. (contains gains, storage variables, ...)
- Initialisation function. (set gains, sample time, ...)
- Update function. (provide setpoint and measurement, returns controller output)