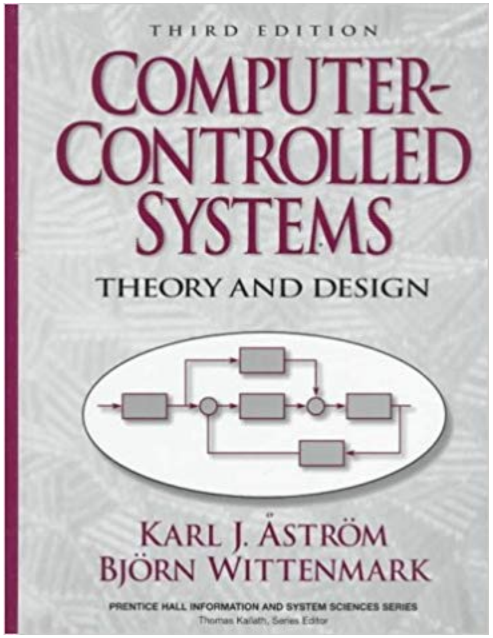
Distributed Real-Time Control Systems

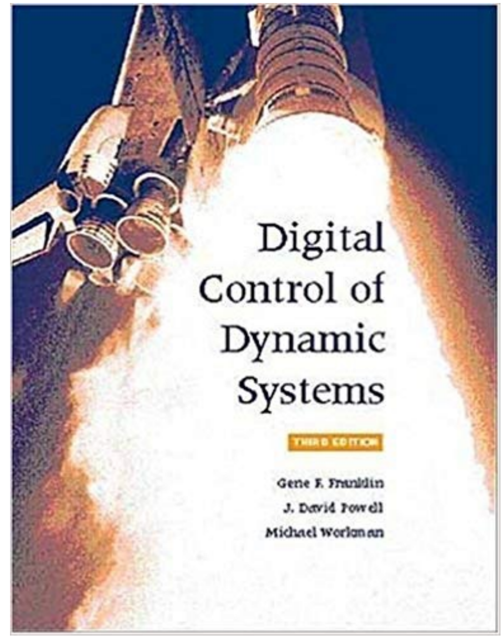
Module 6

PID Control

Bibliography



Section 8.5.



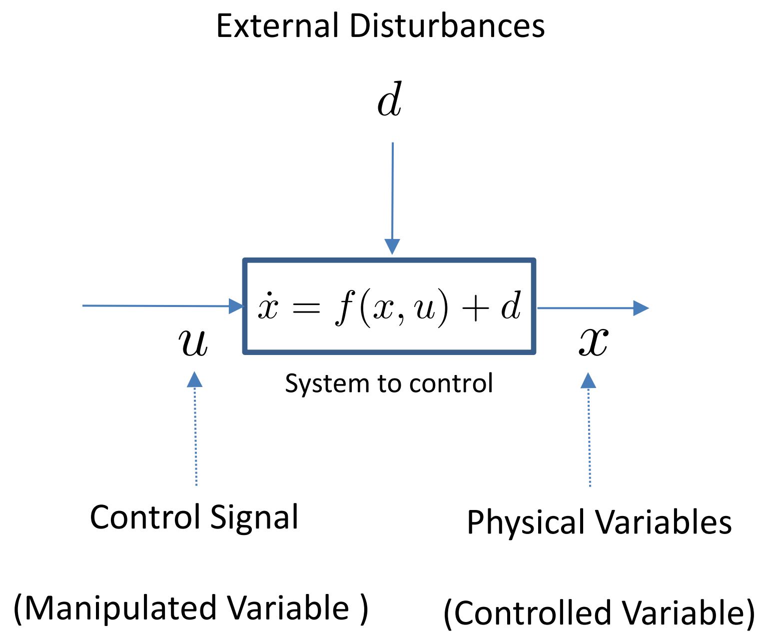
Section 3.3.

Online:

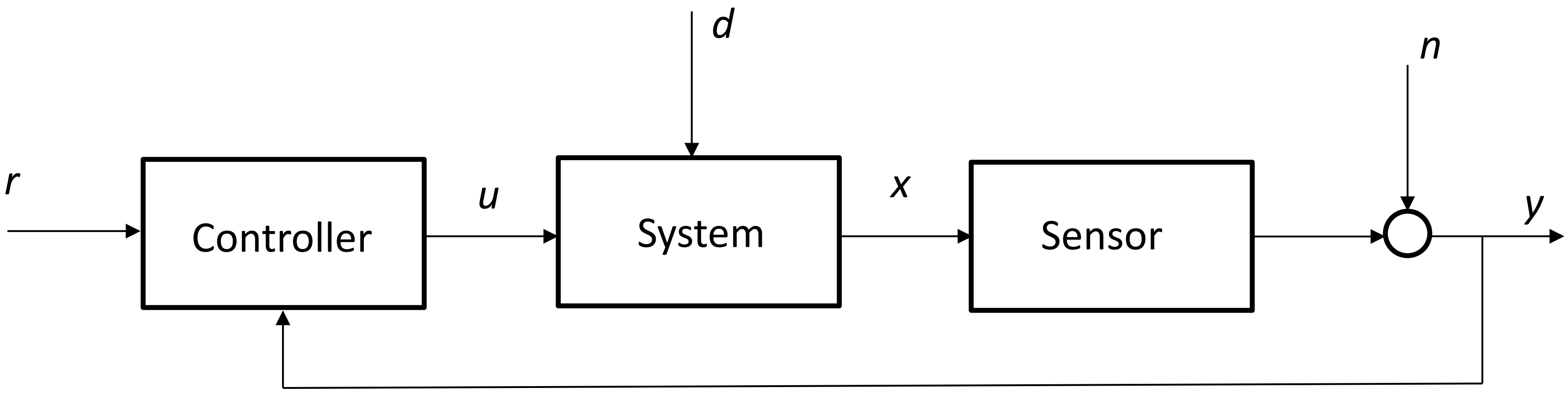
https://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf

Objectives of Control

- Set point control – set the controlled variable to a pre-specified set point.
- Trajectory tracking – make the controlled variable follow a pre-specified trajectory.
- Disturbance rejection – make the controlled variable react to external disturbances to accomplish its task as well as possible.



Illumination Control : Objectives



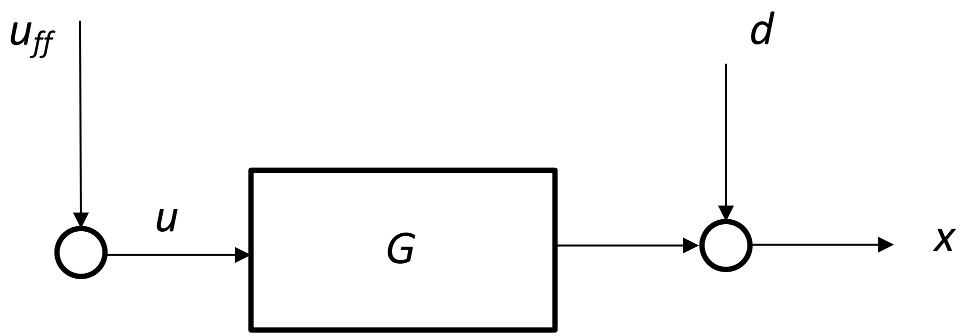
Set point control: set the illuminance to its desired value (*r*) as soon as fast as possible.

Disturbance rejection: keep the value at the desired set point (or above) in reaction to external disturbances (e.g. a window is opened).

In any case, avoid perceivable overshoots and oscillations – these are quite uncomfortable to the users.

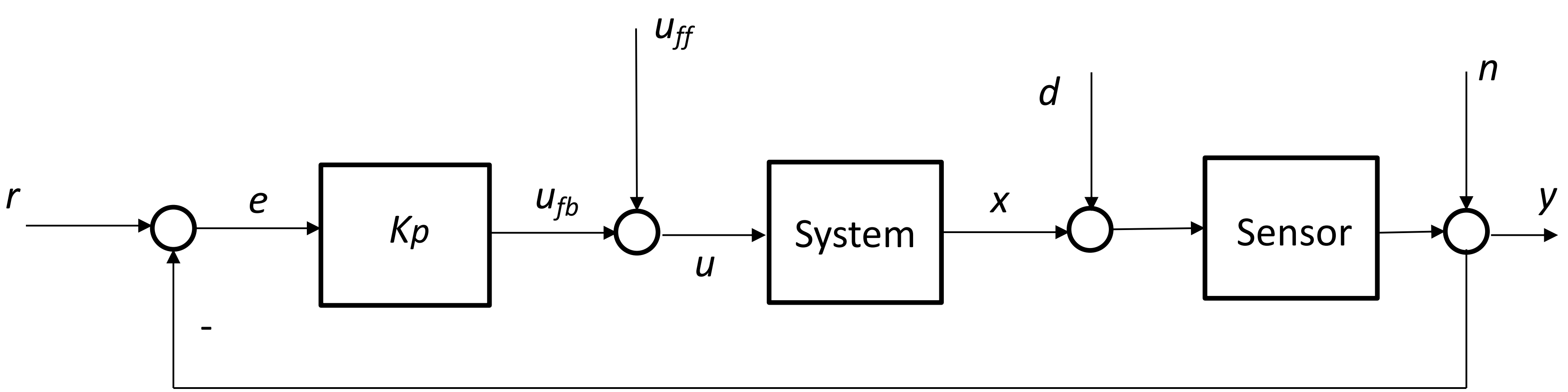
Feedforward Control

- Goal: Set the illuminance to a desired value (*L*)
- Solution – Feedforward control
- Assumption – We know the static gain of the system (G).
- What should be the value of the feedforward command (u_{ff}) ?
- What will happen if the gain G is different from expected ?
- What will happen if there is a non-zero disturbance (d) ?



Proportional Feedback Control

- Goal: React to disturbance **d** (e.g. a window is opened).
- Solution – Proportional control.
- Will the steady state illuminance be the one required ?



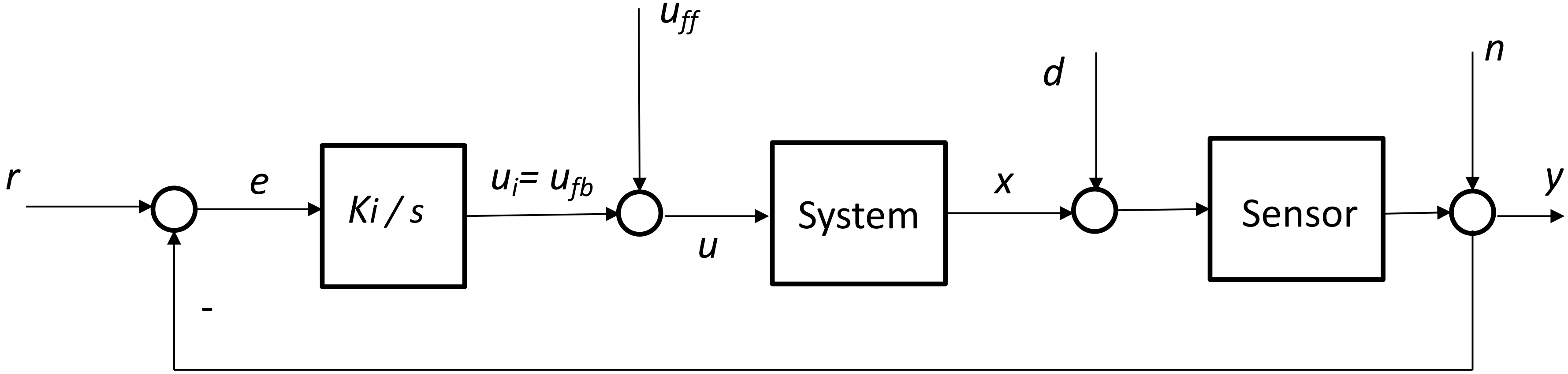
$$u_{fb}(t) = K_p e(t) \qquad \text{with} \qquad e(t) = r(t) - y(t)$$

Integral Feedback Control

Goal: Cancel steady state external disturbances **d** and allow exact set point control.

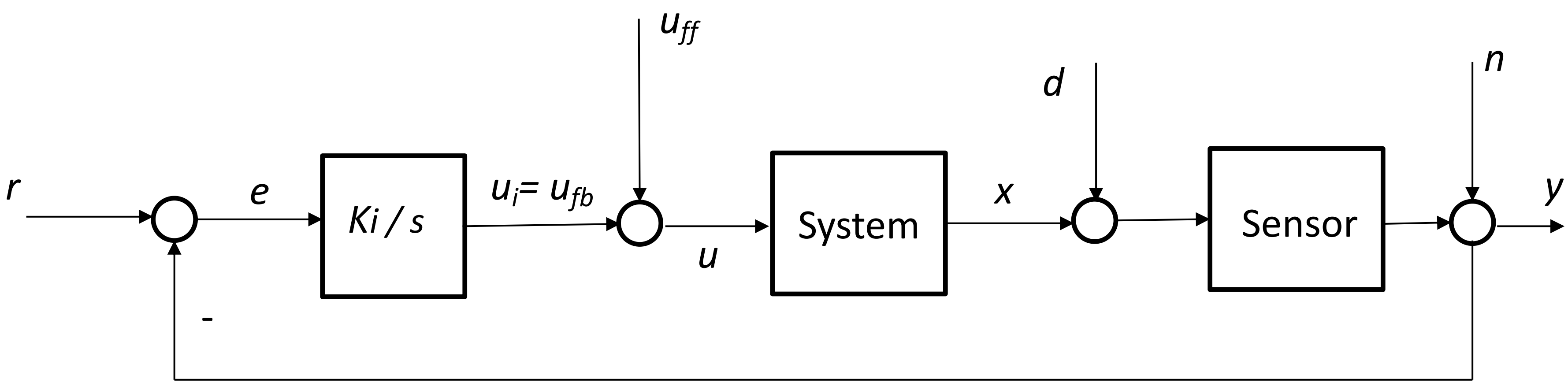
Solution – Proportional Integral control.

What will be the value of the integral term in steady state ?



$$u_{fb}(t) = K_i \int_{t_0}^{t} e(\tau) d\tau$$

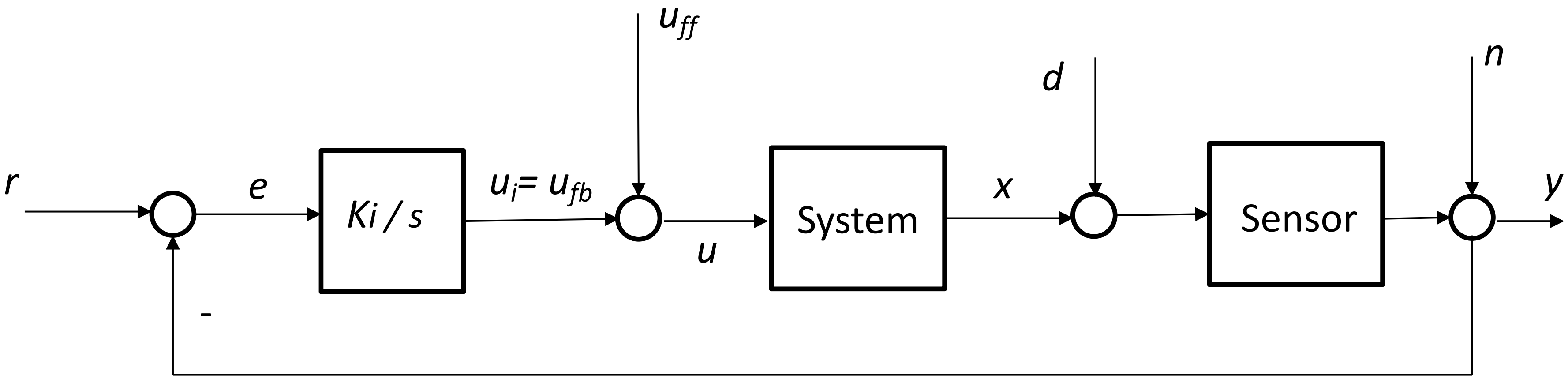
Steady State Analysis



$$u_{fb}(t) = K_i \int_{t_0}^{t} e(\tau) d\tau$$

Show that, if the closed loop is stable, then: $\lim_{t \to \infty} e(t) = 0$

Type 0 systems



Let G be the static gain:

$$y(\infty) = Gu_{ff}(\infty) + Gu_{fb}(\infty) + d(\infty)$$

If the feedforward term is computed as: $u_{ff}(\infty) = G^{-1}y(\infty)$

then the feedback term converges to: $u_{fb}(\infty) = -G^{-1}d(\infty)$

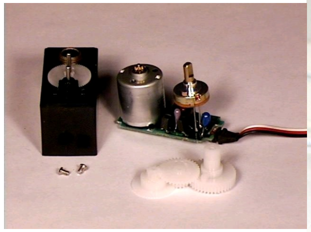
PID Controllers

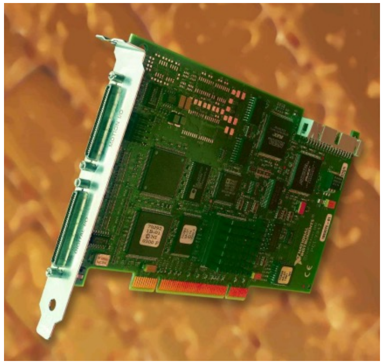
The oldest controllers in industry and still the most used.

Available in most control boards, microcontrollers, automation modules, etc.

In general, can solve 90% dof the control problems: motors, pressure, level, temperature, etc.

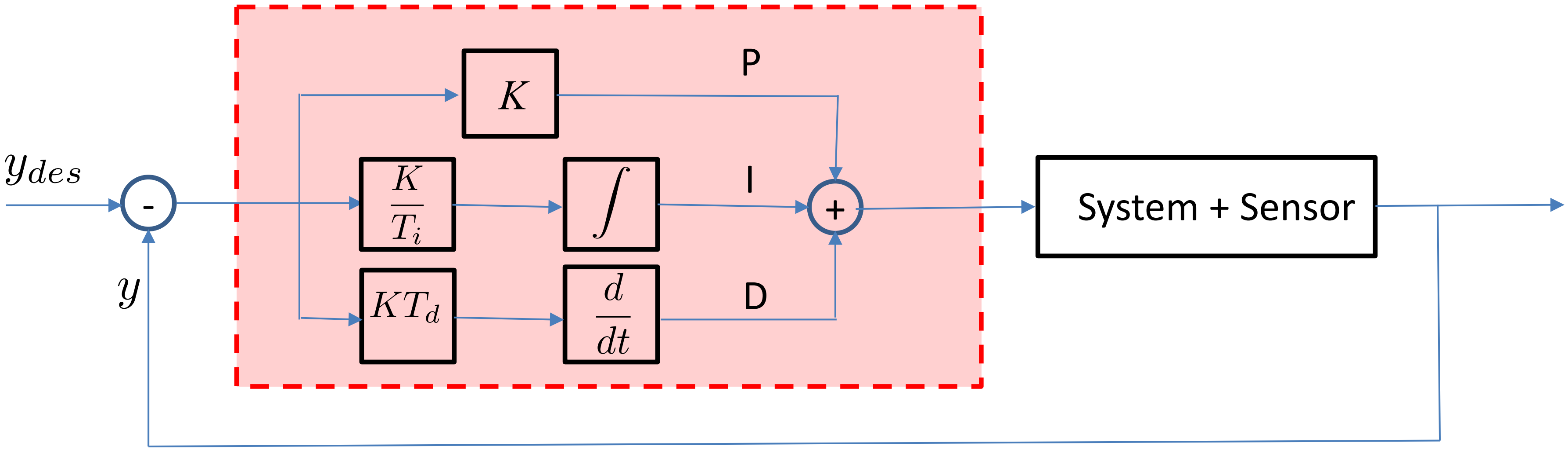
There are particular techniques for their implementation, both in continuous time and discrete time.





PID Control Textbook Version (only feedback)

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



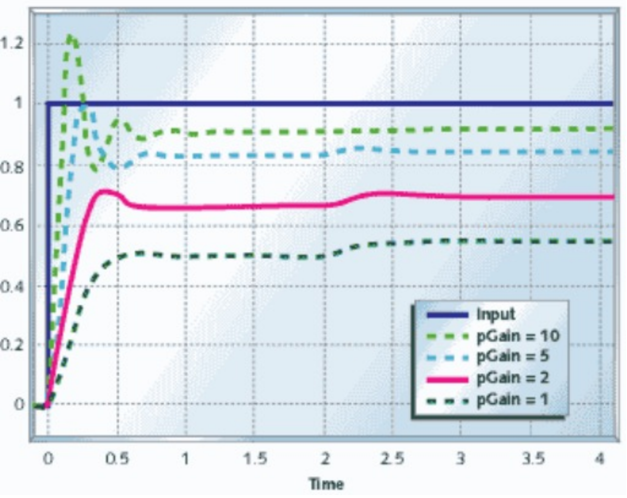
$$U(s) = K\left(1 + \frac{1}{T_i s} + T_d s\right)$$

P - roportional **I - ntegral** **D - erivative**

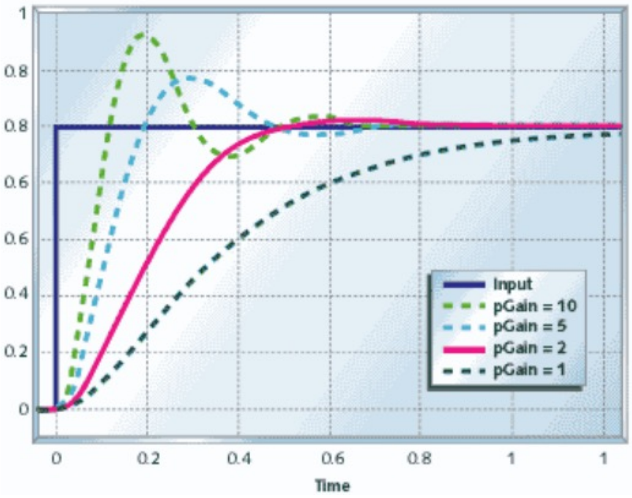
Proportional Term

- Makes reaction faster but:
  - Increases overshoot and the frequency of oscillations
  - May not lead to zero static error (for systems without integrators).
  - May not stabilize the system (for systems with significant delays).

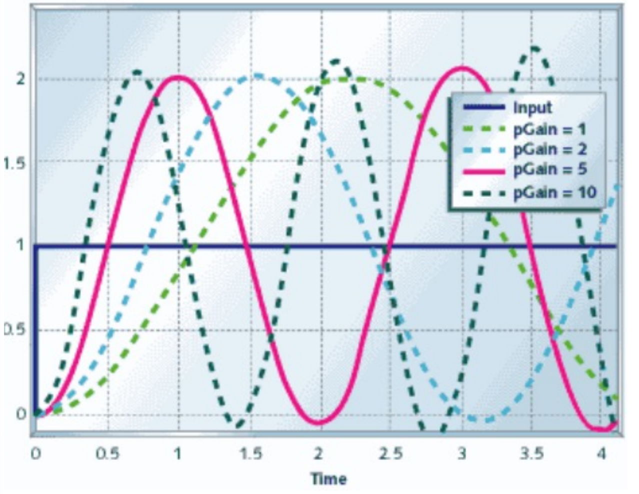
Type 0 system



Type 1 system

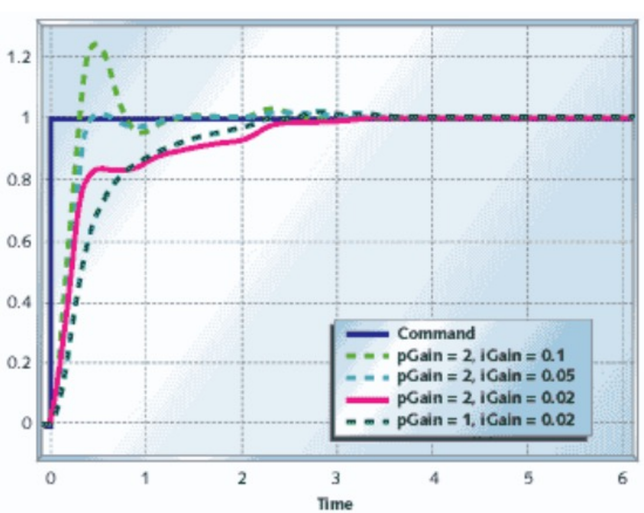


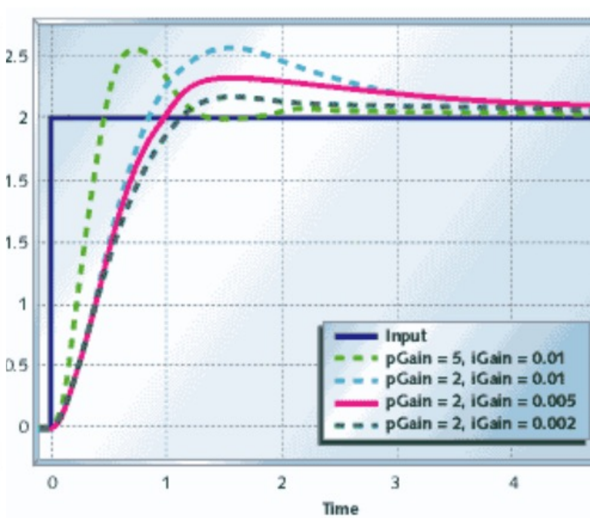
System with significant delay

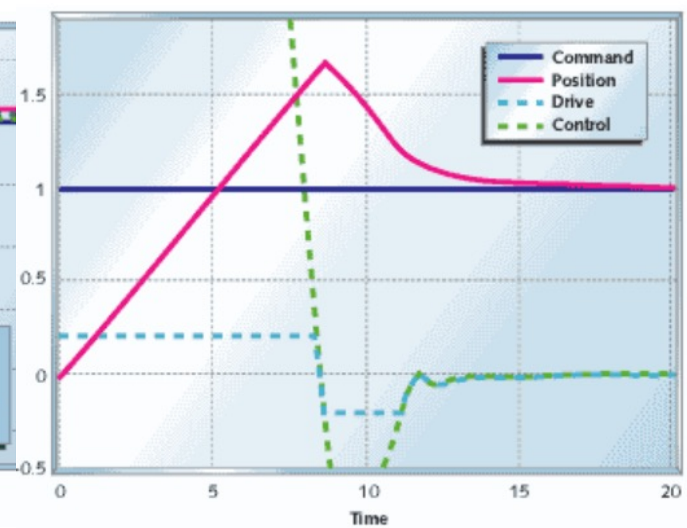


Integral Term

- Integral control alone decreases stability. It is usually combined with proportional control.
  - PI control is effective for systems without integrators (achieves steady state zero error and makes response faster).
  - PI control may slow down systems with integrators.
  - PI control may wind-up if actuators saturate.

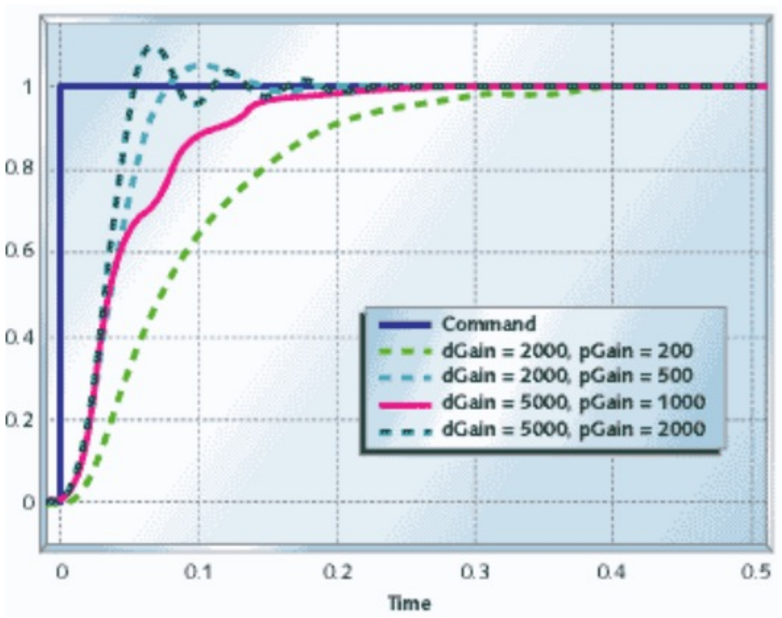


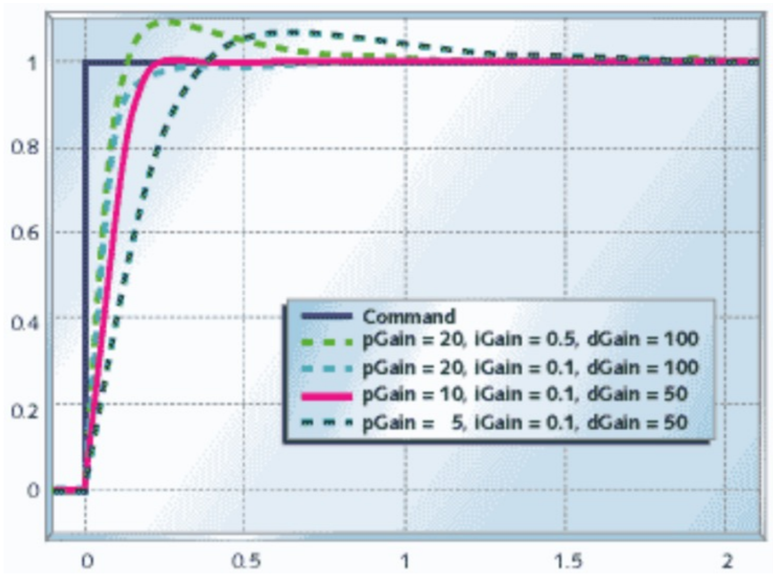




Derivative Control

- Derivative terms do some prediction about the output evolution.
  - PD control is effective in stabilizing systems with large delays.
  - PID control is able to speed up response and keep low overshoot and oscillations frequency.
  - However, is very sensitive to noise.

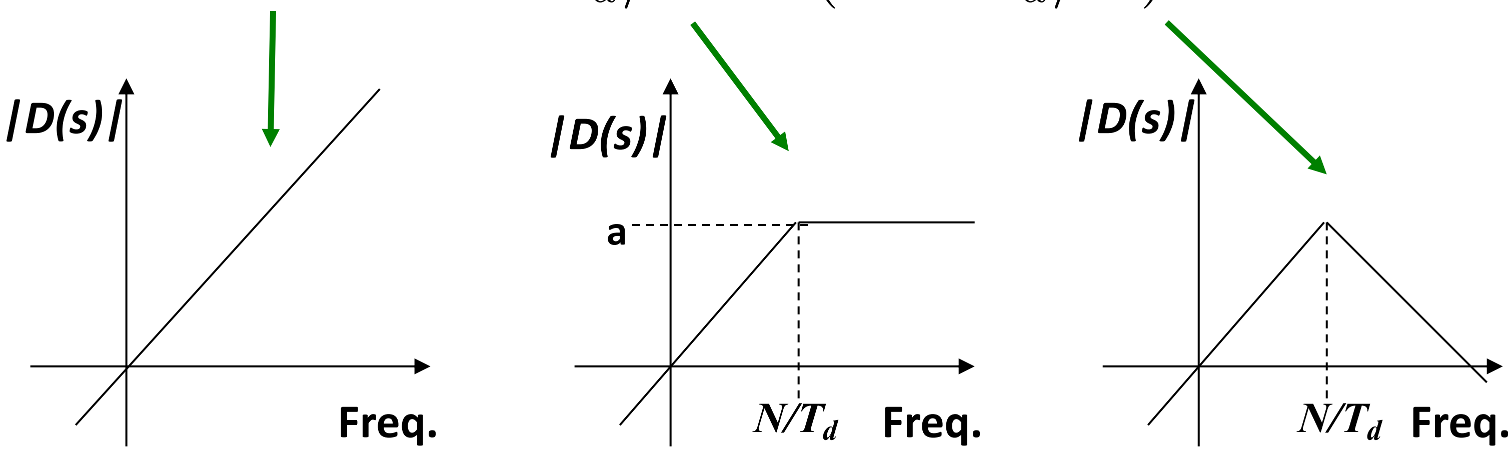




Derivative Term w/ Noise Reduction

The derivative operator should not be applied "as is" to very high frequency signals, such as noise and discontinuities in the error signal. Additional low-pass terms, with bandwidth ***N***, are introduced:

$$sT_d \approx \frac{sNT_d}{1 + sT_d/N} \approx \frac{sNT_d}{(1 + sT_d/N)^2}$$



Note: for the range of frequencies of interest *]0,0.1a/T_d]* the three operators are equivalent i.e. they have the same gain and phase.

Typical values for *N* in [3…20]

Set Point Weighting

The Textbook version of the PID controller has two main problems:

- Does not include feedforward term to improve set-point control
- The derivative term is subject to step changes in the reference (what happens when a derivative sees a discontinuity?)

The **PID with Set Point Weighting** addresses these problems.

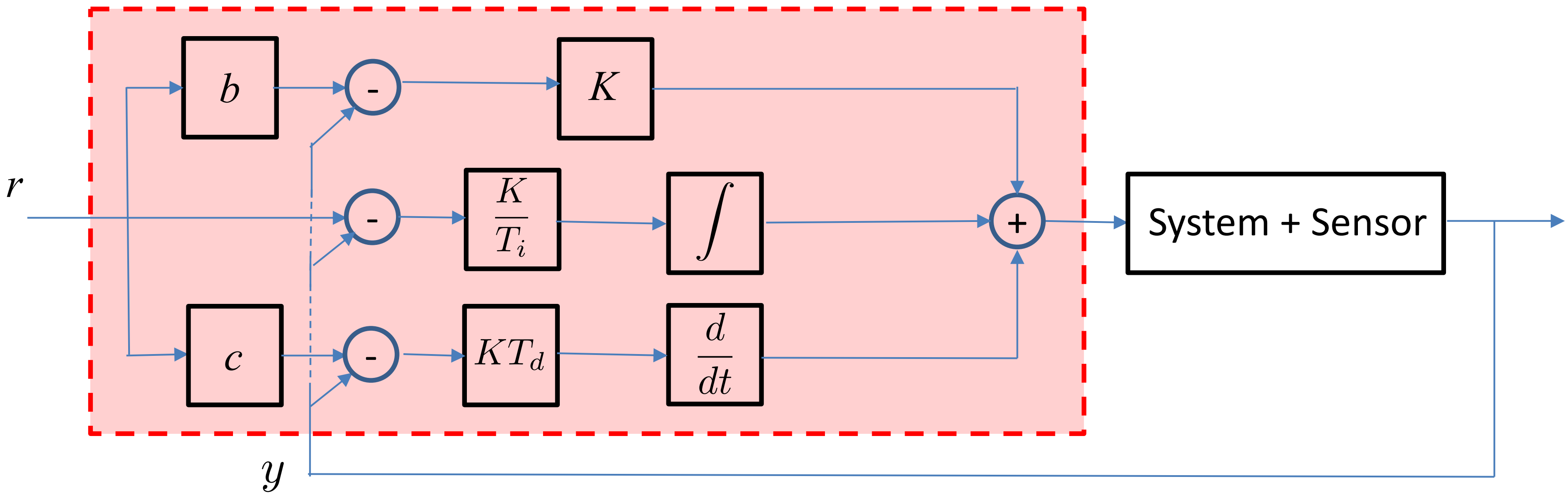
$$u(t) = K\left(br(t) - y(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\left(c\frac{dr(t)}{dt} - \frac{dy(t)}{dt}\right)\right)$$

Compare to the textbook form:

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

Set Point Weighting

$$u(t) = K\left(br(t) - y(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\left(c\frac{dr(t)}{dt} - \frac{dy(t)}{dt}\right)\right)$$



Set Point Weighting

$$u(t) = K\left(br(t) - y(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\left(c\frac{dr(t)}{dt} - \frac{dy(t)}{dt}\right)\right)$$

The PID controller with set point weighting is a **controller with two degrees of freedom** because it decouples the transfer functions Y -> U and R -> U

$$\frac{U(s)}{Y(s)} = C_y(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right)$$

Identical to original

$$\frac{U(s)}{R(s)} = C_r(s) = K\left(b + \frac{1}{sT_i} + csT_d\right)$$

Can tune b and c to improve set point control

Model-Based vs Model-Free PID Tuning

Model-Based Tuning

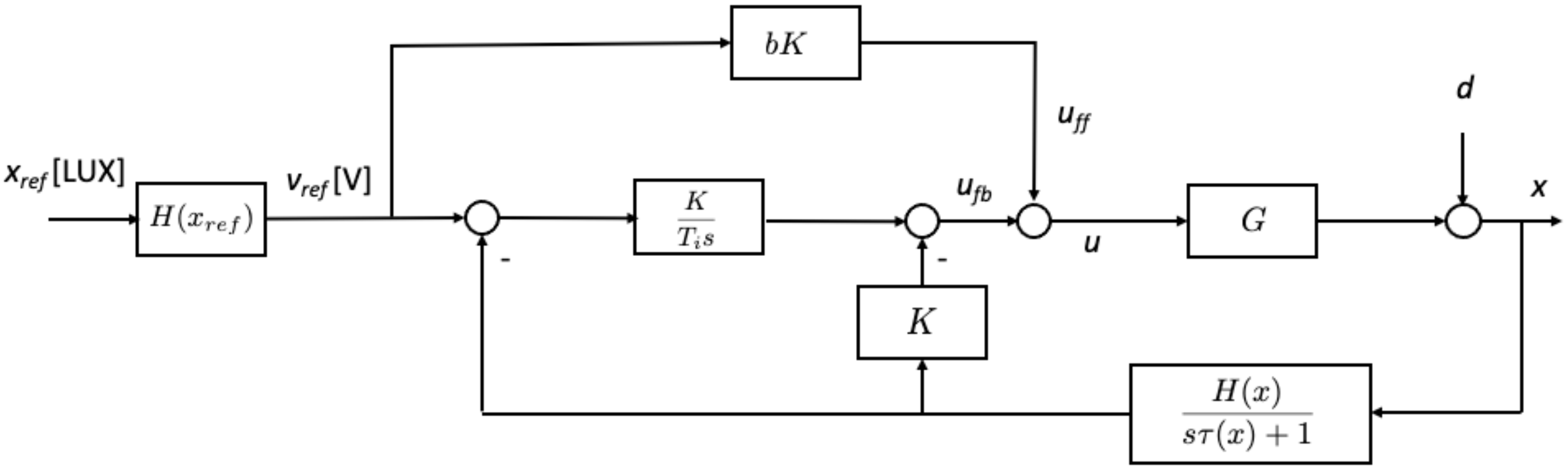
- Use a model of the system (or an approximation) to analytically compute the properties of the controlled systems as a function of the controller parameters.

Model-Free Tuning

- Tune the controller parameters on the real system by trial and error. Some tips in https://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf, section 6.6.

Case Study: Project System

Analysis of a set point weighting PID controller for the linear approximation of the lab system (check slides of last lecture to understand the involved approximations and assumptions).



Compute the transfer function from x_{ref} to x (set-point TF) and from d to x (disturbance TF)

Set-point TF

$$X = G\left(bKHX_{ref} - \frac{KH}{s\tau+1}X + \frac{K}{T_i s}(HX_{ref} - \frac{H}{s\tau+1}X)\right)$$

$$X\left(1 + \frac{GHK}{s\tau+1} + \frac{GHK}{(s\tau+1)T_i s} + \right) = X_{ref}\left(bGHK + \frac{GHK}{T_i s}\right)$$

$$X\left(\frac{(s\tau+1)T_i s + GHKT_i s + GHK}{(s\tau+1)T_i s} + \right) = X_{ref}\left(\frac{bGHKT_i s + GHK}{T_i s}\right)$$

$$\frac{X}{X_{ref}} = \frac{(bGHKT_i s + GHK)(s\tau+1)}{(s\tau+1)T_i s + GHKT_i s + GHK}$$

$$\frac{X}{X_{ref}} = \frac{(bT_i s + 1)(s\tau+1)}{(s\tau+1)\frac{T_i s}{GHK} + T_i s + 1}$$

Set-point TF

$$\boxed{\frac{X}{X_{ref}} = \frac{(bT_i s + 1)(s\tau + 1)}{(s\tau + 1)\frac{T_i s}{GHK} + T_i s + 1}}$$

Homework:

1. Show that the resulting illuminance converges to the reference value.
2. Compute the disturbance TF
3. Select good values for the PID controller: b, Ti, K

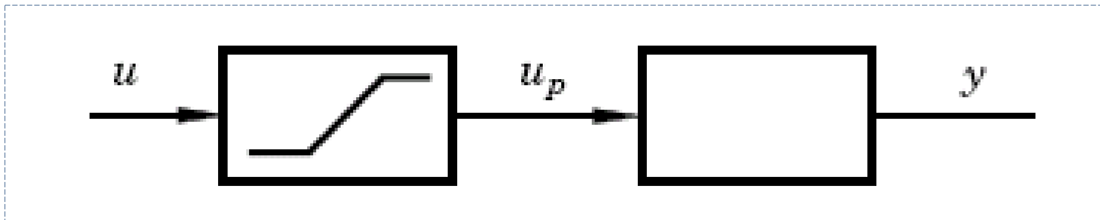
Advice for Project

Note that the project system is non-linear: gain and time constant depend on the illuminance. The PID gains may need to be dynamically changed depending on the set points.

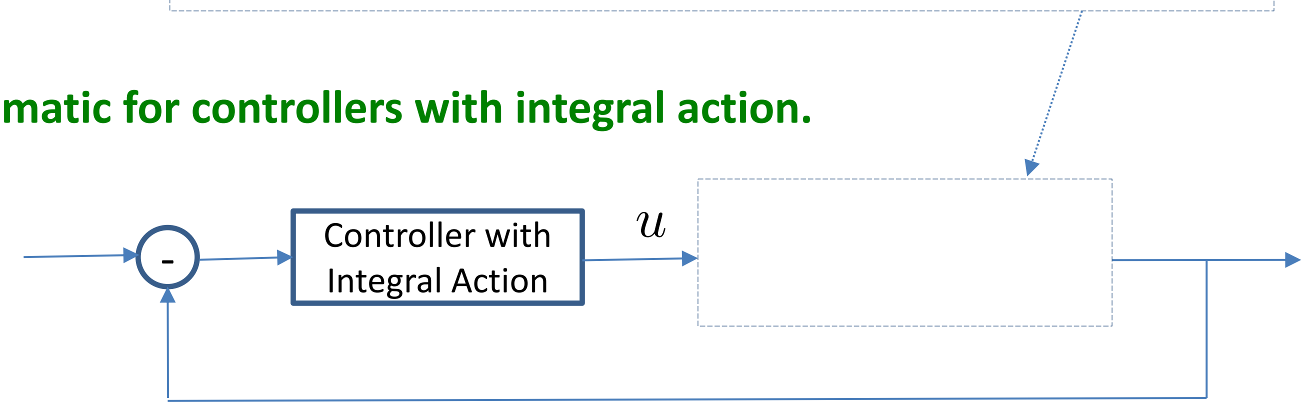
Because the approximation is non-ideal, you may still need to fine tune the controller weights. But you should be able to get a good starting point.

Saturations in the Control Action

Every actuator has limits!



Problematic for controllers with integral action.



When the control signal saturates, the integral part keeps integrating. Known as *integrator-windup* or *reset-windup.***

**an integrator is often denoted by "reset" due to its effect in the controller. When appropriately used, it "resets" the tracking errors.

Windup-Effect

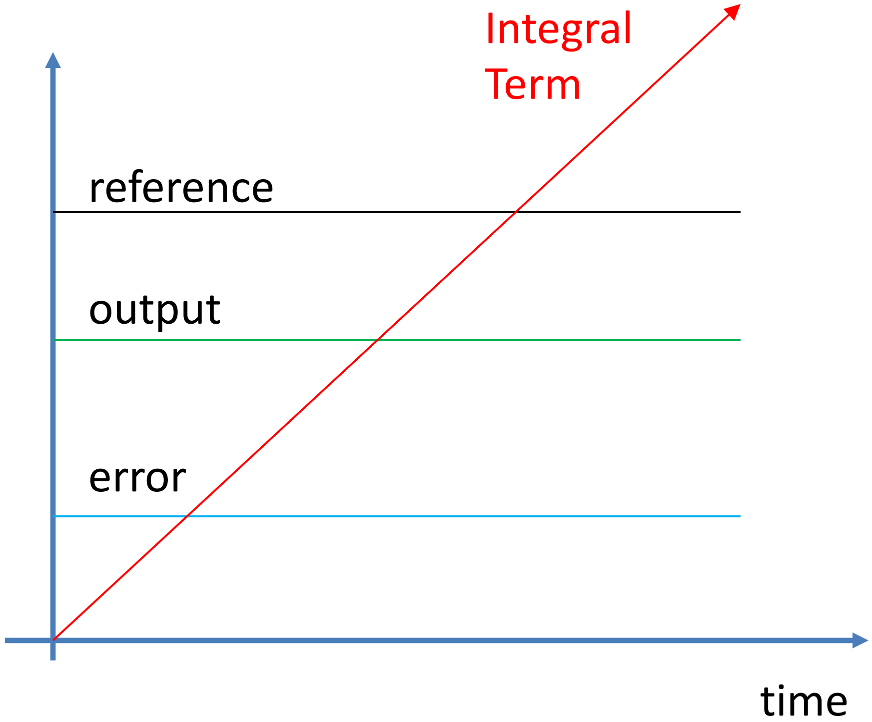
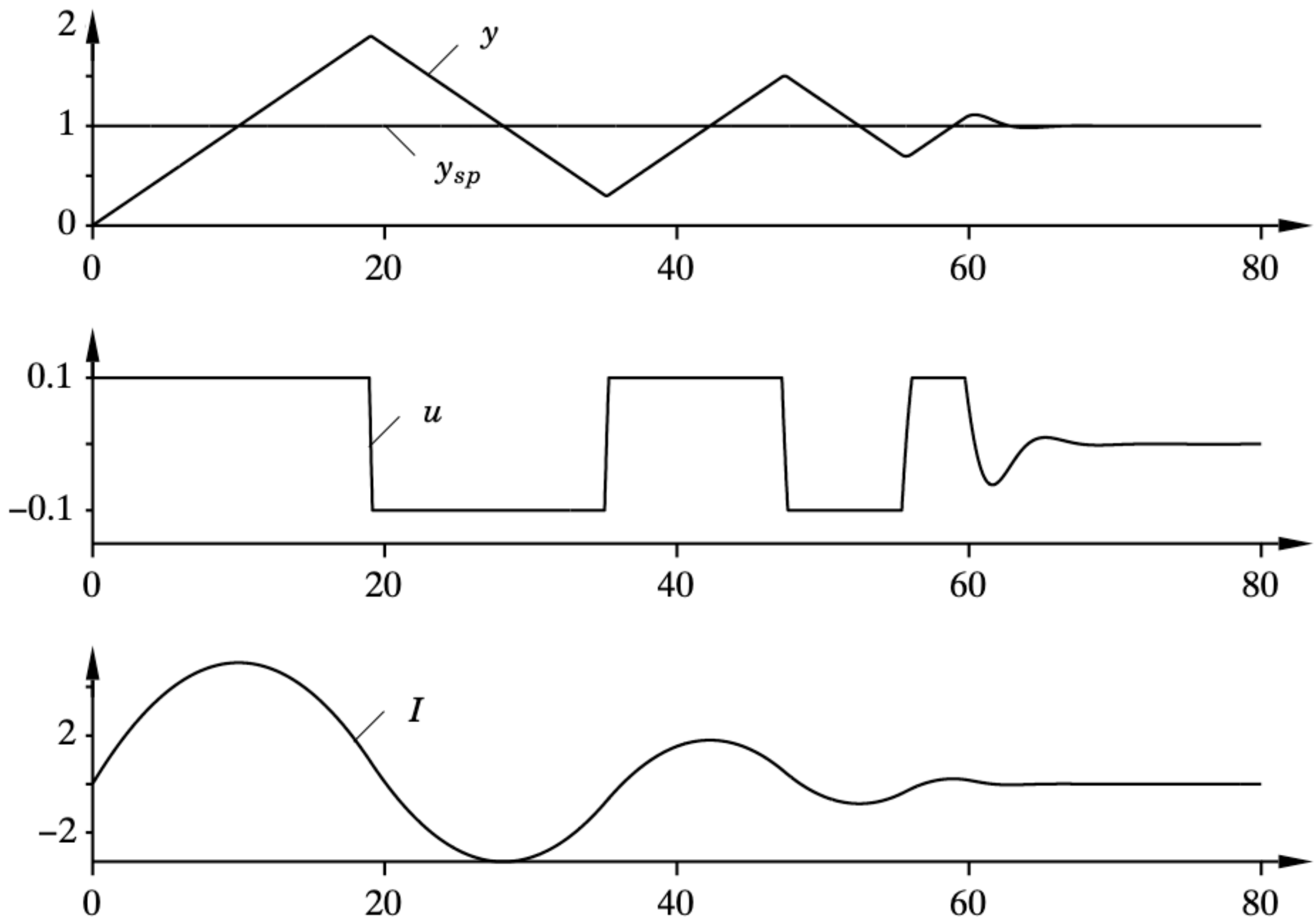


Illustration of the Windup Effect

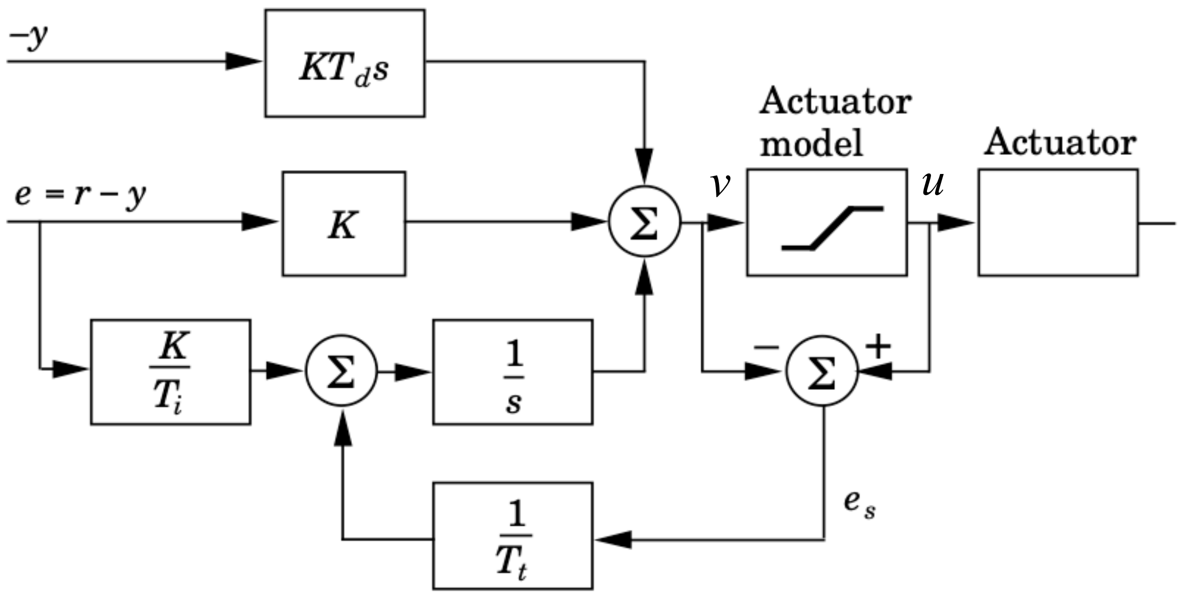


Anti-Windup

Some Windup solutions:

- **Setpoint and Setpoint rate limitation**
  - **Slows down system response.**
  - **Does not prevent windup due to disturbances.**
- **Reset integral term**
  - **Generates discontinuities in the command.**
- **Stop Integrating.**
  - **Only discharges the integrator when actuator is not saturated.**
- **Back-Calculation (slow discharge integrator)**
  - **Must tune one extra parameter.**

Back Calculation



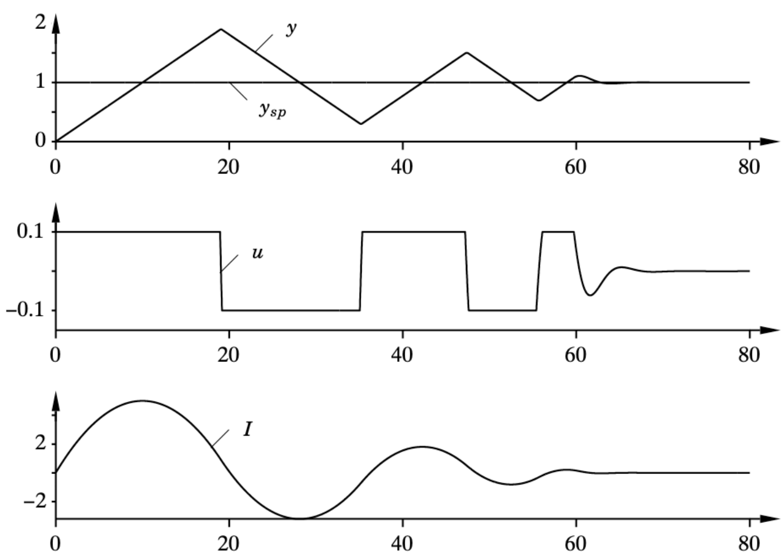
Saturation error $e_s = u - v$

During the saturation period, the saturation error e_s tends to $$e_s = -\frac{KT_t}{T_i} e$$

In steady state the output tends to $$v = u_{\lim} + \frac{KT_t}{T_i} e$$

Back Calculation

Before



After