Embedded and Real-time Systems

Control

http://d3s.mff.cuni.cz

Department of Distributed and Dependable Systems

D3S

Tomáš Bureš

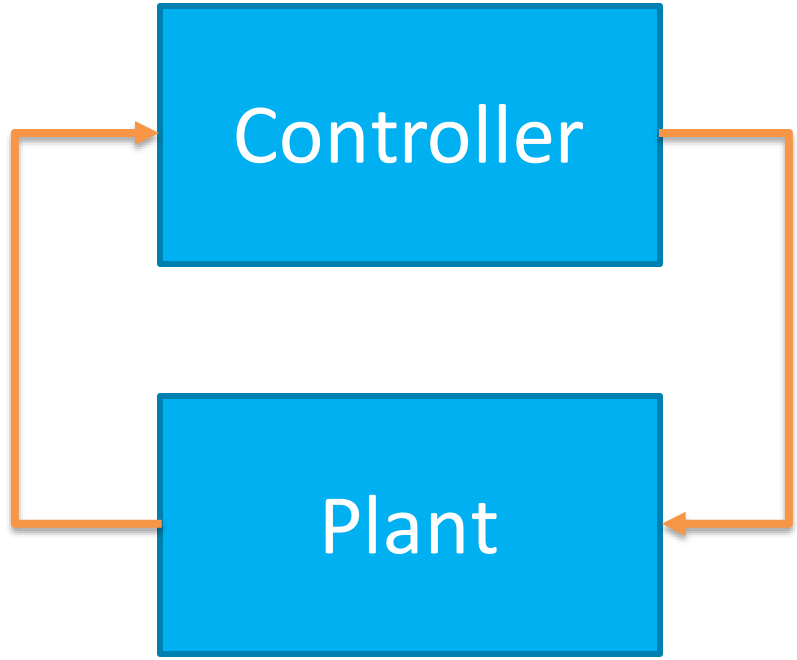
<bures@d3s.mff.cuni.cz>

CHARLES UNIVERSITY IN PRAGUE

Faculty of Mathematics and Physics

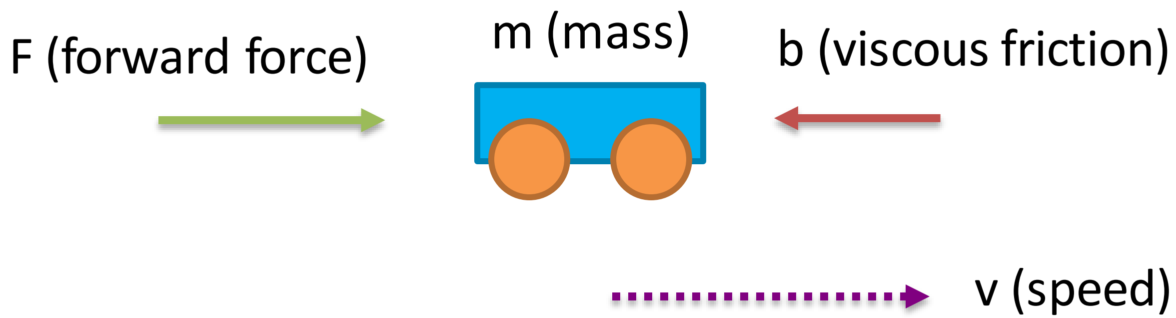
Controlling Environment

- Plant
  - continuous dynamic system
- Controller
  - discrete reactive system



- Controller brings plant to a desired state and keeps it there
- Controller keeps a setpoint (the desired state)

Example: moving body



$$\frac{\partial v}{\partial t} = \frac{F - bv}{m}$$

Task: regulate F to accelerate to a given speed and keep the speed

Modeling using block diagram

- Block is a function
  - Input/output ports
  - Some blocks may have internal state (e.g. integrator)
- Connections
  - Data flow between ports

- Simulink (an extension to MATLAB) to model a system using blocks
  - Simulation, analysis, code generation

- Example: moving_body_1_no_controller.slx

Simple relay control

- Switch maximum force on/off
  - Example: moving_body_2_relay.slx

- Fails when there is a lag in the system
  - e.g. because of delayed actuation or because of periodic sampling/actuation
  - Example: moving_body_3_lag_relay.slx

PID Controller

- Typical controller for linear state space systems
  - Linear state-space systems is a system that can be described by the following differential equation:

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx + Du$$

- Example (moving body): $\frac{\partial v}{\partial t} = \frac{F - bv}{m}$

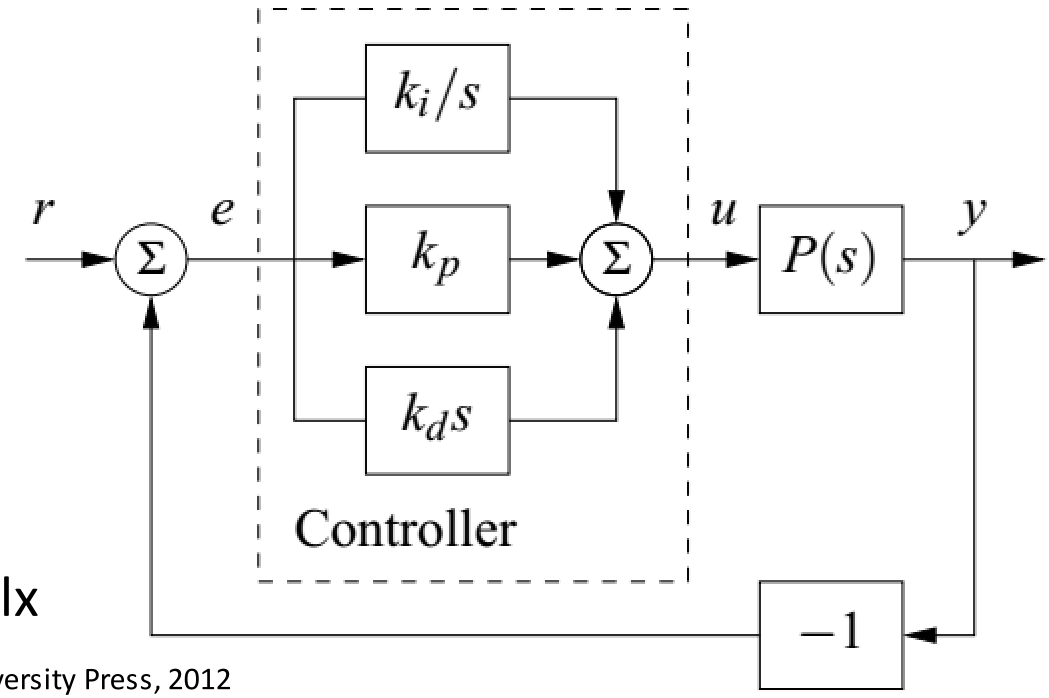
$$\frac{dx}{dt} = -\frac{b}{m}x + \frac{1}{m}F, \quad v = 1x + 0F$$

PID Controller

- Ideal form:

$$u = k_p e + k_i \int_0^t e(\tau) d\tau + k_d \frac{de}{dt}$$

- Weighted sum of three terms:
  - Proportional
  - Integral
  - Differential



- Example: moving_body_4_pi.slx

Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Proportional term

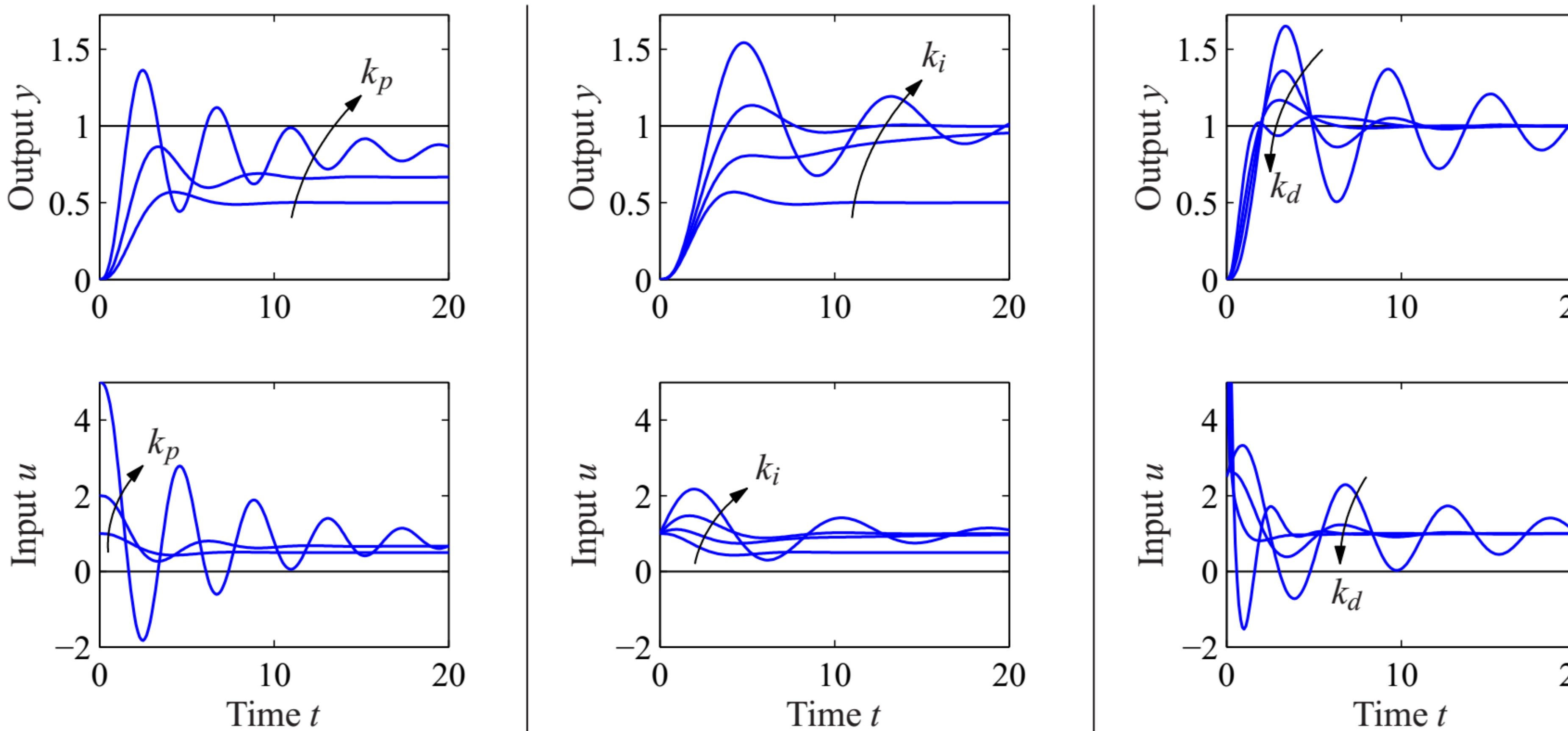
- Counter-acts proportionally to the error

- Low k_p
  - Slow action
- High k_p
  - May overshoot and oscillate

- Problem
  - Never reaches the set-point if there is a steady resistance
  - Can be mitigated by an extra feedforward term, but this term may vary with the internal state of the system
    - e.g. the viscous friction

- Example: vehicle_speed_1_p.slx

Effect of constants in PID



(a) Proportional control (b) PI control (c) PID control

Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Integral term

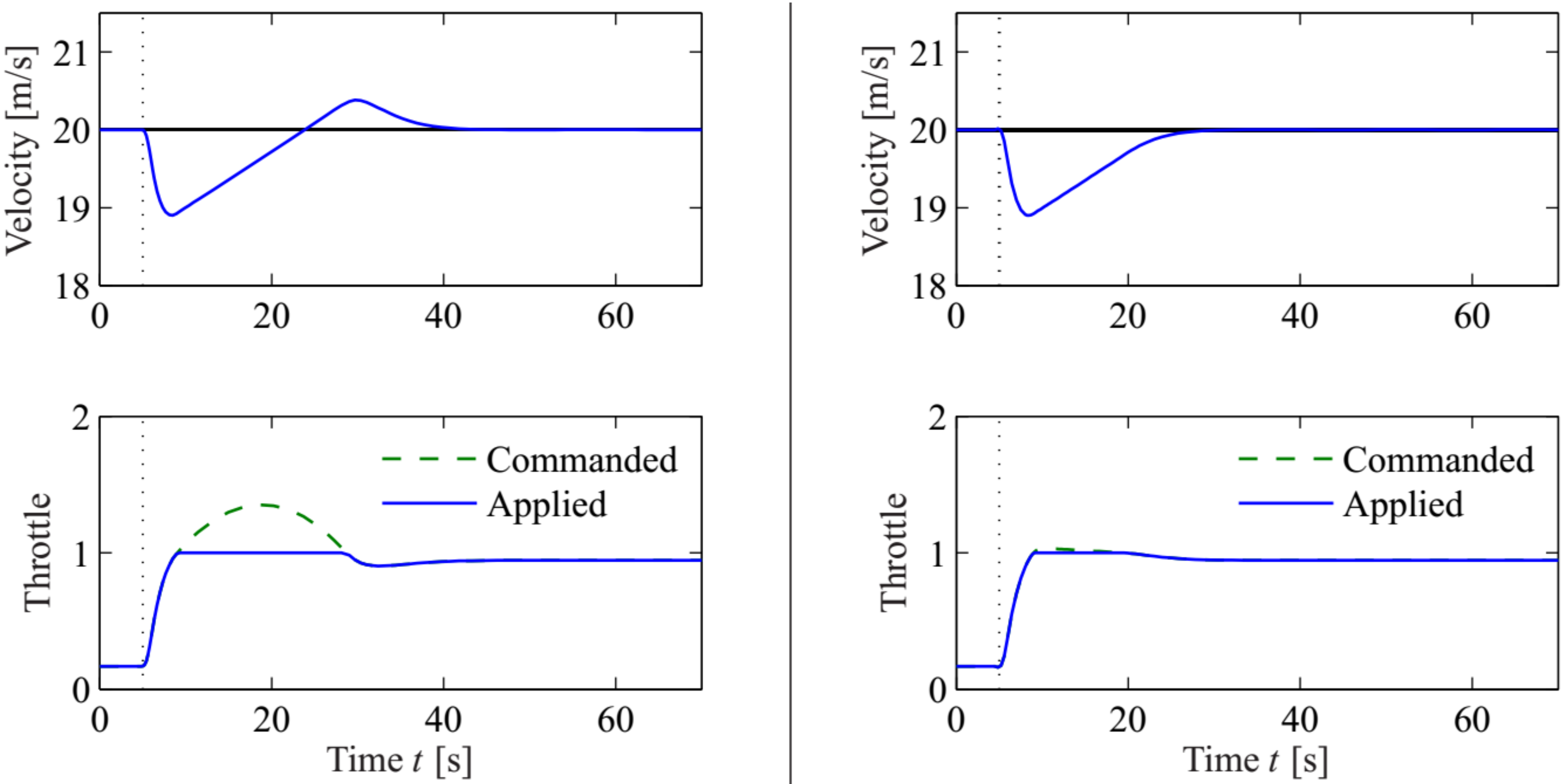
- Mitigates the steady-state error

- Low k_i
  - Slow gradual “learning”
- High k_i
  - Big overshoot and oscillation

- Example: vehicle_speed_2_pi.slx
  - Note that pure I controller would work too, just slower

- Problem – integrator windup
  - Happens when actuator reaches the saturation limit
  - Integrator mistakenly accumulates value
  - Example: vehicle_speed_3_pi_windup.slx

Integrator windup



(a) Windup

(b) Anti-windup

Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Anti-windup

- Prevents the integrator to accumulate when saturation is reached
- Example: vehicle_speed_4_pi_anti_windup.slx

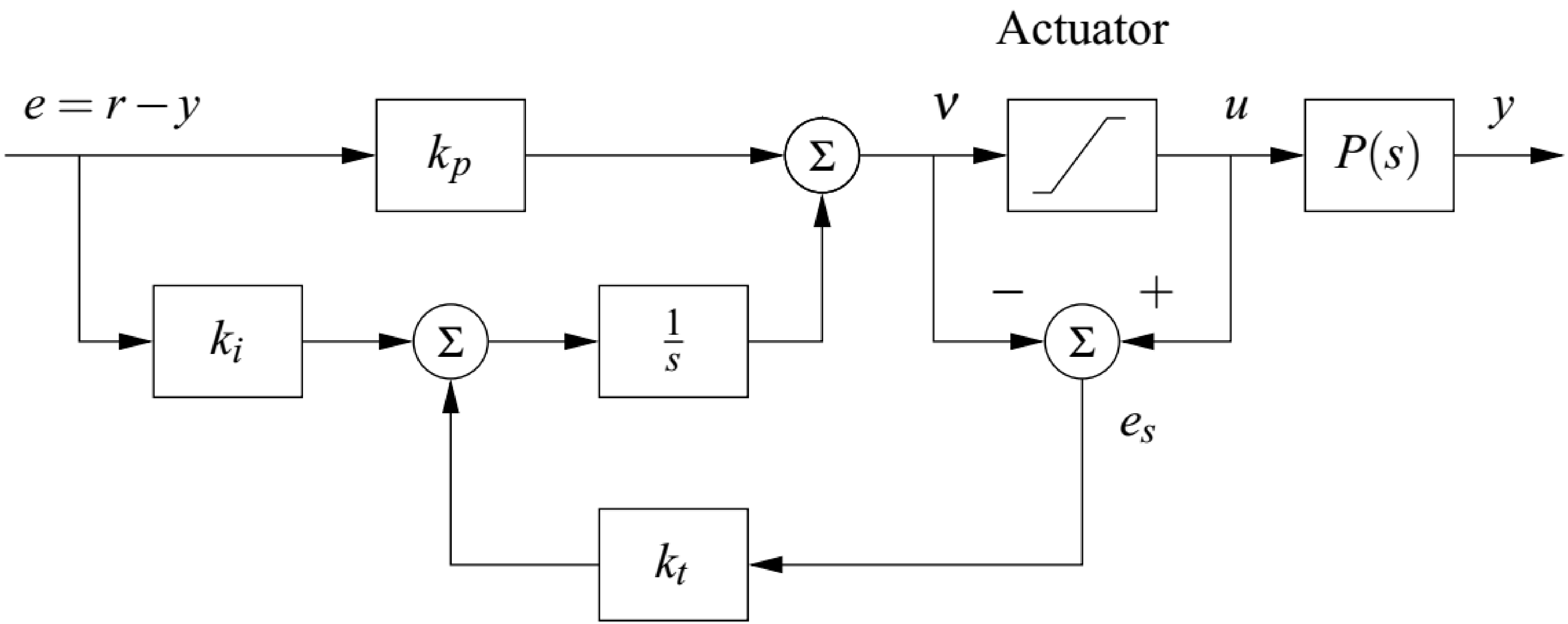


Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Transfer functions

- Differential and integral blocks described using transfer function $G(s)$

- $G(s) = \frac{u}{y}$
  - u … output from the controller / input to the plant
  - y … output from the plant

- Transfer function describes the effect on frequency and phase of a periodic signal
  - $|G(i\omega)|$ … gain
  - $\angle G(i\omega)$ … phase shift

Transfer functions – Bode plot

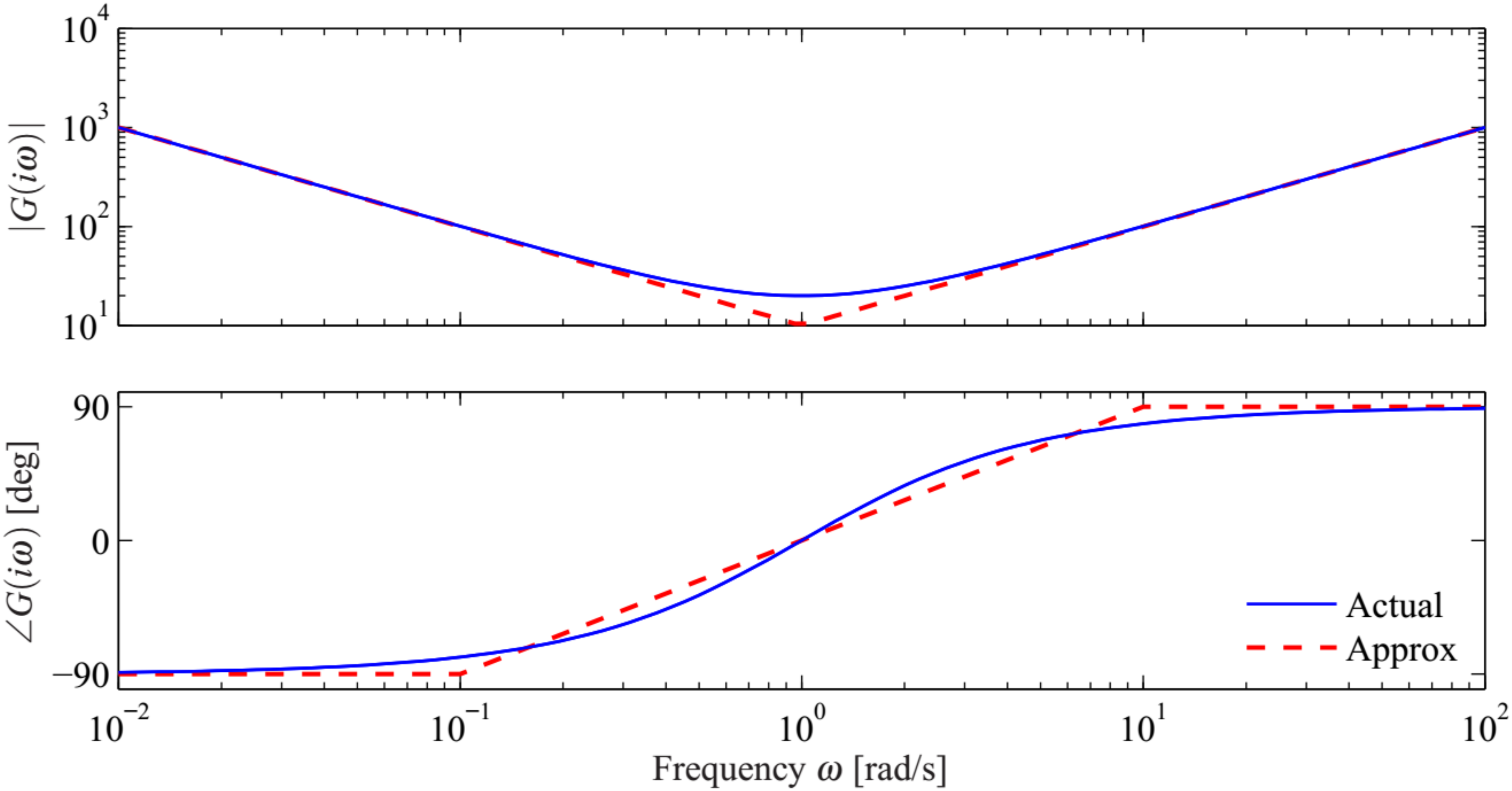


Figure 8.11: Bode plot of the transfer function $C(s) = 20 + 10/s + 10s$ corresponding to an ideal PID controller. The top plot is the gain curve and the bottom plot is the phase curve.

Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Transfer functions – examples

Type	ODE	Transfer Function
Integrator	$\dot{y} = u$	$\dfrac{1}{s}$
Differentiator	$y = \dot{u}$	s
First-order system	$\dot{y} + ay = u$	$\dfrac{1}{s+a}$
Double integrator	$\ddot{y} = u$	$\dfrac{1}{s^2}$
Damped oscillator	$\ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2 y = u$	$\dfrac{1}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$
PID controller	$y = k_p u + k_d \dot{u} + k_i \int u$	$k_p + k_d s + \dfrac{k_i}{s}$
Time delay	$y(t) = u(t-\tau)$	$e^{-\tau s}$

Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Variable s

- Exponential (periodic) signal e^{st}
  - $s = \sigma + i\omega$ is a complex variable
  - σ … decay rate (if $\sigma < 0$), ω … frequency

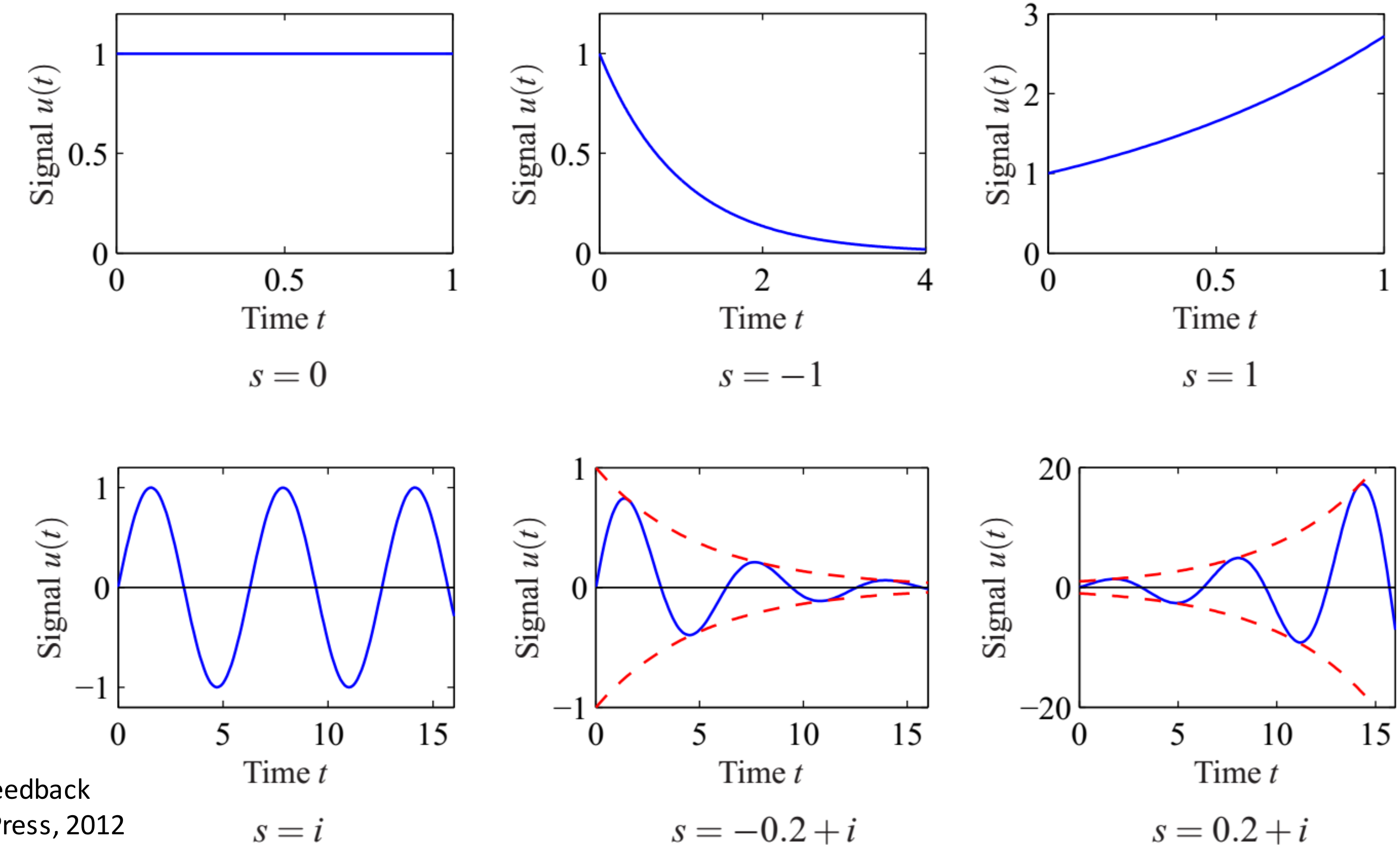
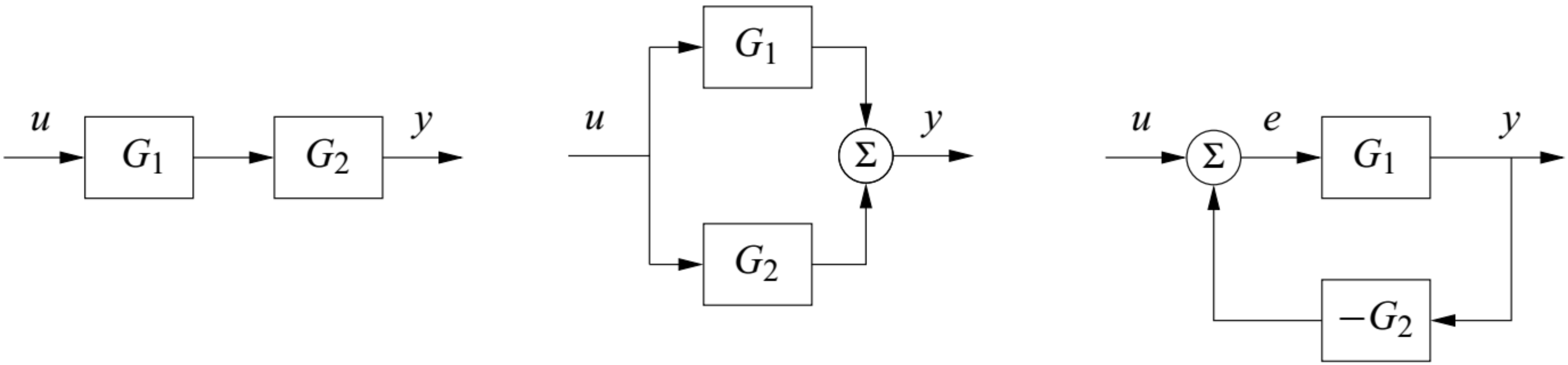


Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Transfer functions – block diagrams



(a) $G_{yu} = G_2 G_1$

(b) $G_{yu} = G_1 + G_2$

(c) $G_{yu} = \dfrac{G_1}{1 + G_1 G_2}$

Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Proof of case (c):

$$y = G_1 e,\ e = u - G_2 y$$
$$y = G_1 u - G_1 G_2 y$$
$$y(1 + G_1 G_2) = G_1 u$$

$$y = \frac{G_1}{1 + G_1 G_2} u$$

Derivative term

- Provides prediction of error in the future
  - Does linear extrapolation
- Reduces oscillations and overshoot

- Low k_d
  - Small effect on oscillations
- High k_d
  - Reduces controller response
  - May itself create oscillations

- Examples: vehicle_position_1_p.slx, vehicle_position_2_pd.slx, vehicle_position_3_pid.slx

- Problem – sensitivity to high frequencies (noise)
  - Derivative term amplifies the high frequencies
  - Mitigated by low-pass filtering

Department of Distributed and Dependable Systems

Effect of constants in PID

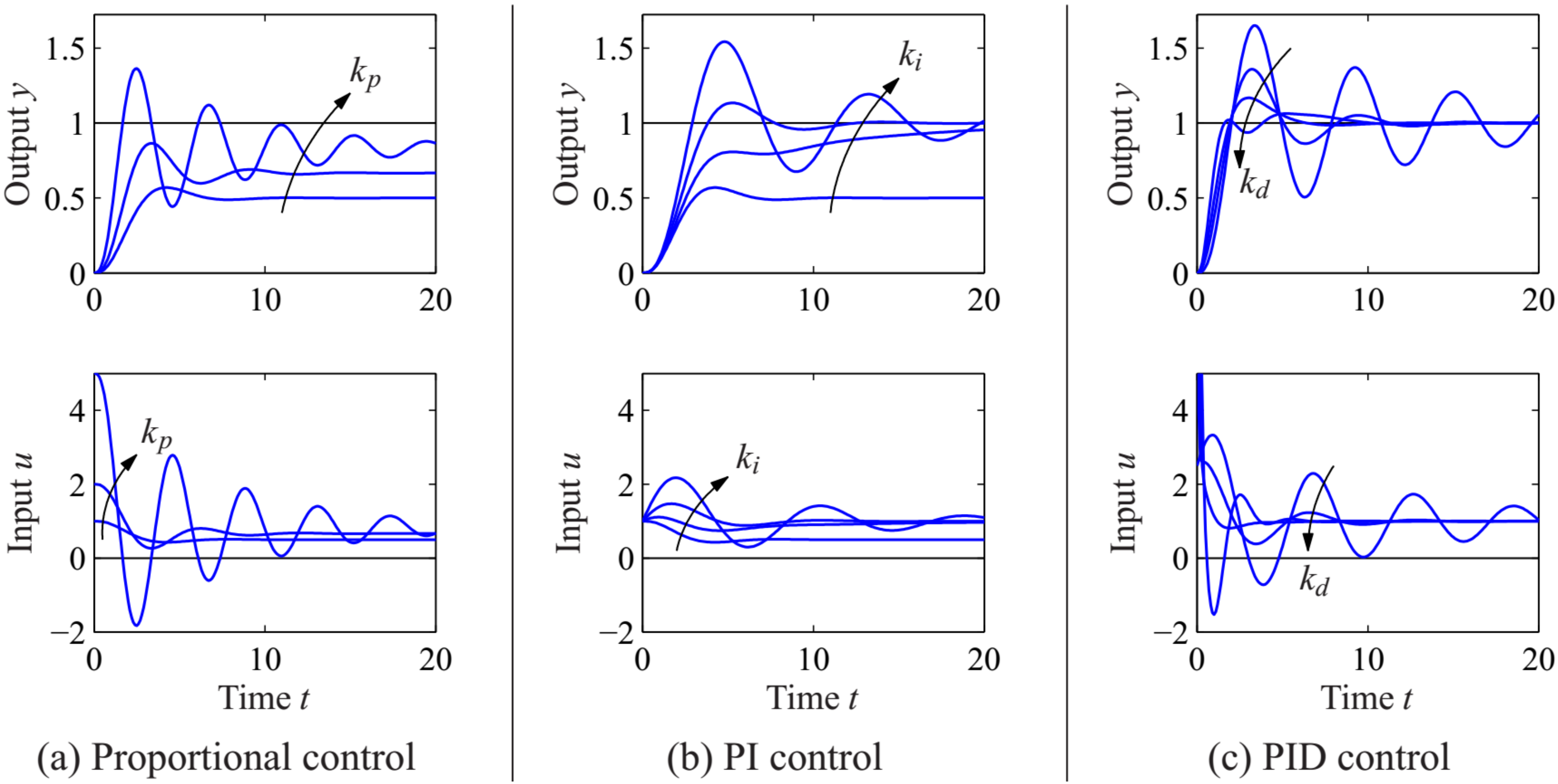


Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Filtered derivative

- Derivative term $D = k_d s$ replaced by $D = k_d \frac{s}{1+T_f s}$
  - for small frequencies acts as derivative
  - for high frequencies acts as a constant gain

- To mitigate spike when setpoint r is changed, it can take $-y$ as the input (instead of $e = r - y$)
  - For constant setpoint, the computation is the same because r as a constant gets discarded in the differential

Department of Distributed and Dependable Systems

Filtered derivative

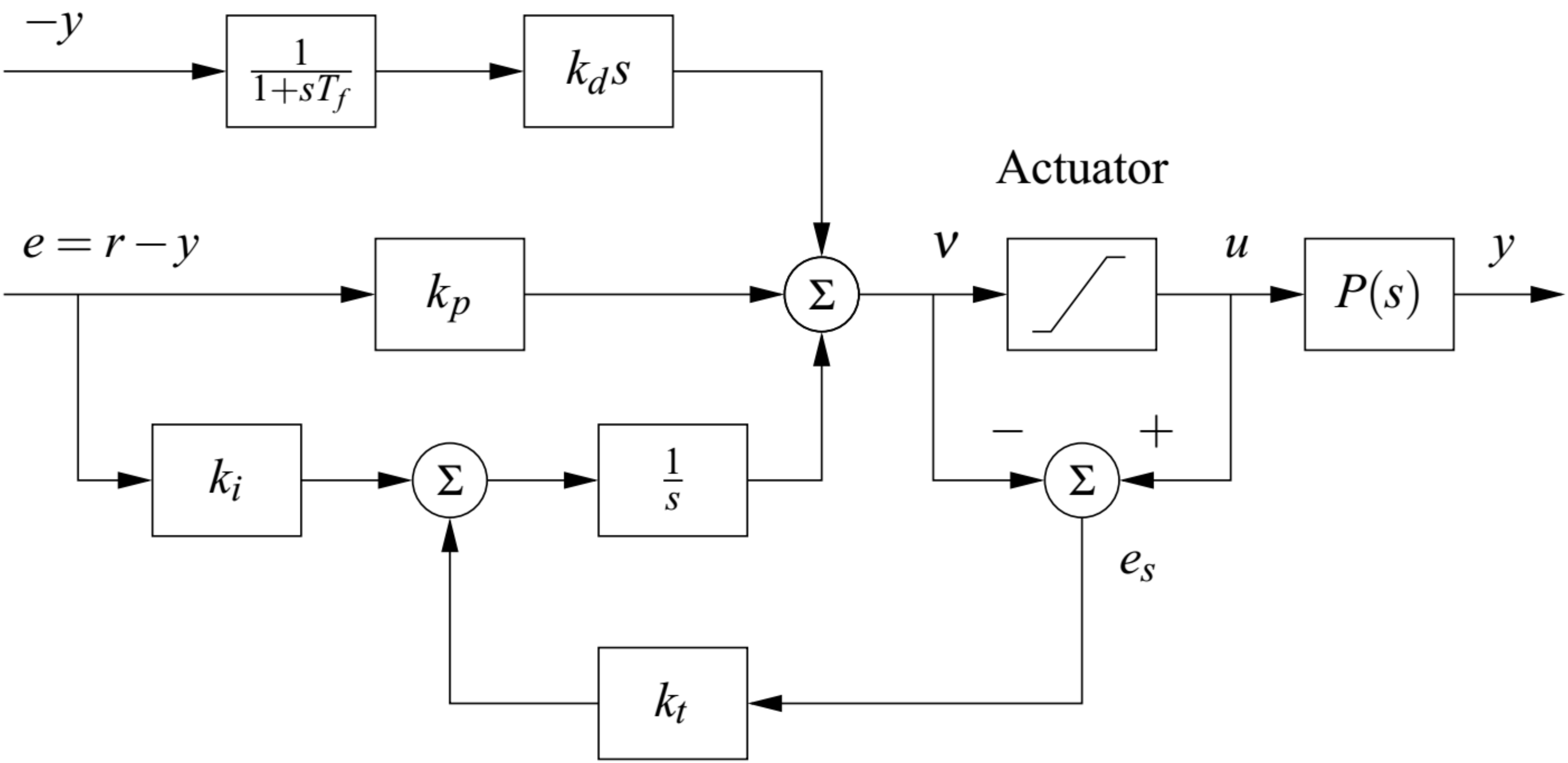


Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

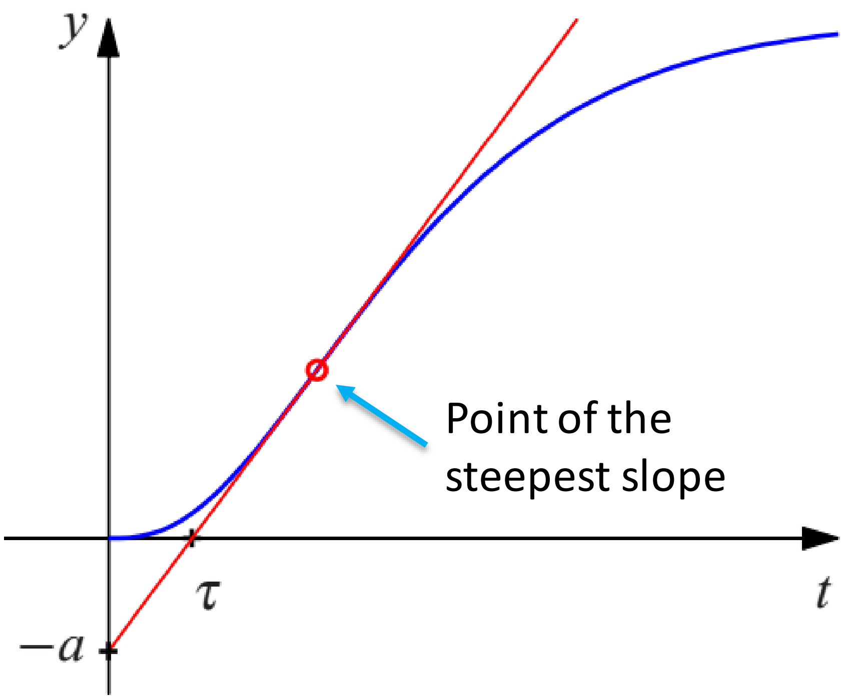
Tuning

- Different methods for initial estimation of the constants k_p, k_i, k_d
- Manual fine-tuning may be required

Ziegler-Nichols step response method

Unit step is applied

and response measured



Constants for controller

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

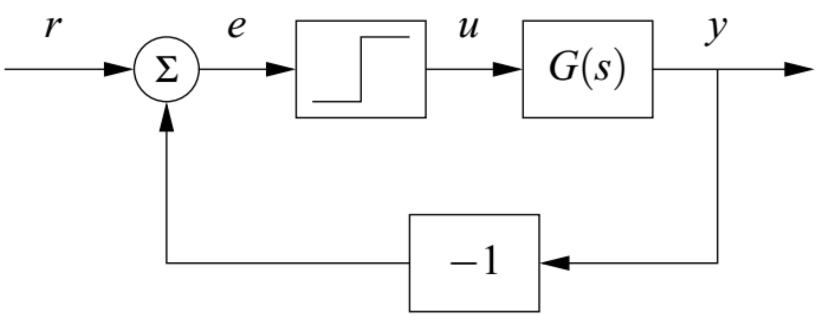
computed as:

Type	k_p	T_i	T_d
P	1/a		
PI	0.9/a	3τ	
PID	1.2/a	2τ	0.5τ

Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Ziegler-Nichols frequency response

Using relay feedback to bring the system to oscillation



Oscillatory response

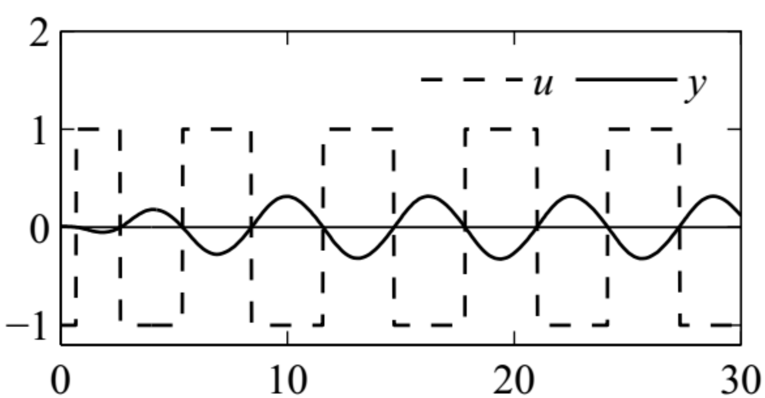


Figure from Åström, Murray: Feedback Systems, Princeton University Press, 2012

Constants for controller

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

computed as:

Type	k_p	T_i	T_d
P	$0.5k_c$		
PI	$0.4k_c$	$0.8T_c$	
PID	$0.6k_c$	$0.5T_c$	$0.125T_c$

where:

T_c … oscillation period
d … relay amplitude
a … process amplitude
$K_c = \frac{4d}{a\pi}$ … critical gain

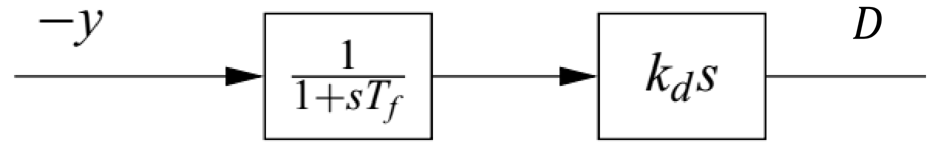
Implementation

- $P(t_k) = k_p\big(r(t_k) - y(t_k)\big)$

- $I(t_{k+1}) = I(t_k) + k_i h e(t_k) + k_t h(\mathrm{sat}(v) - v)$
  - h is the discrete time step

- $D(t_k) = \frac{T_f}{T_f + h} D(t_{k-1}) - \frac{k_d}{T_f + h}\big(y(t_k) - y(t_{k-1})\big)$
  - steps to arrive at D on the next slide

Steps to derive $D(t_k)$



$$D = -\frac{y k_d s}{1 + sT_f}$$

$$D + T_f D s = -k_d y s$$

Applying the transfer function s on the respective terms:

$$T_f \frac{dD}{dt} + D = -k_d \dot{y}$$

Approximating the derivative with backward difference

$$T_f \frac{D(t_k) - D(t_{k-1})}{h} + D(t_k) = -k_d \frac{y(t_k) - y(t_{k-1})}{h}$$

Department of Distributed and Dependable Systems

Pseudo-code

```
% Precompute controller coefficients
bi = ki * h
ad = Tf / (Tf + h)
bd = kd / (Tf + h)
br = kt * h

% Control algorithm - main loop
while (running) {
  r = adin(ch1)                        % read setpoint from ch1
  y = adin(ch2)                        % read process variable from ch2
  P = kp * (r - y)                     % compute proportional part
  D = ad * D - bd * (y - yold)         % update derivative part
  v = P + I + D                        % compute temporary output
  u = sat(v, ulow, uhigh)              % simulate actuator saturation
  daout(ch1)                           % set analog output ch1

  I = I + bi * (r - y) + br * (u - v)  % update integral
  yold = y                             % update old process output

  wait_for_next_period
}
```

Department of
Distributed and
Dependable
Systems

Removing non-linearities

- Sometimes the process has non -linearities
  - E.g. coulomb friction (can be modeled as a relay)
- Can be addressed by conditioning the process
  - E.g. adding compensation to the coulomb friction to output of the controller