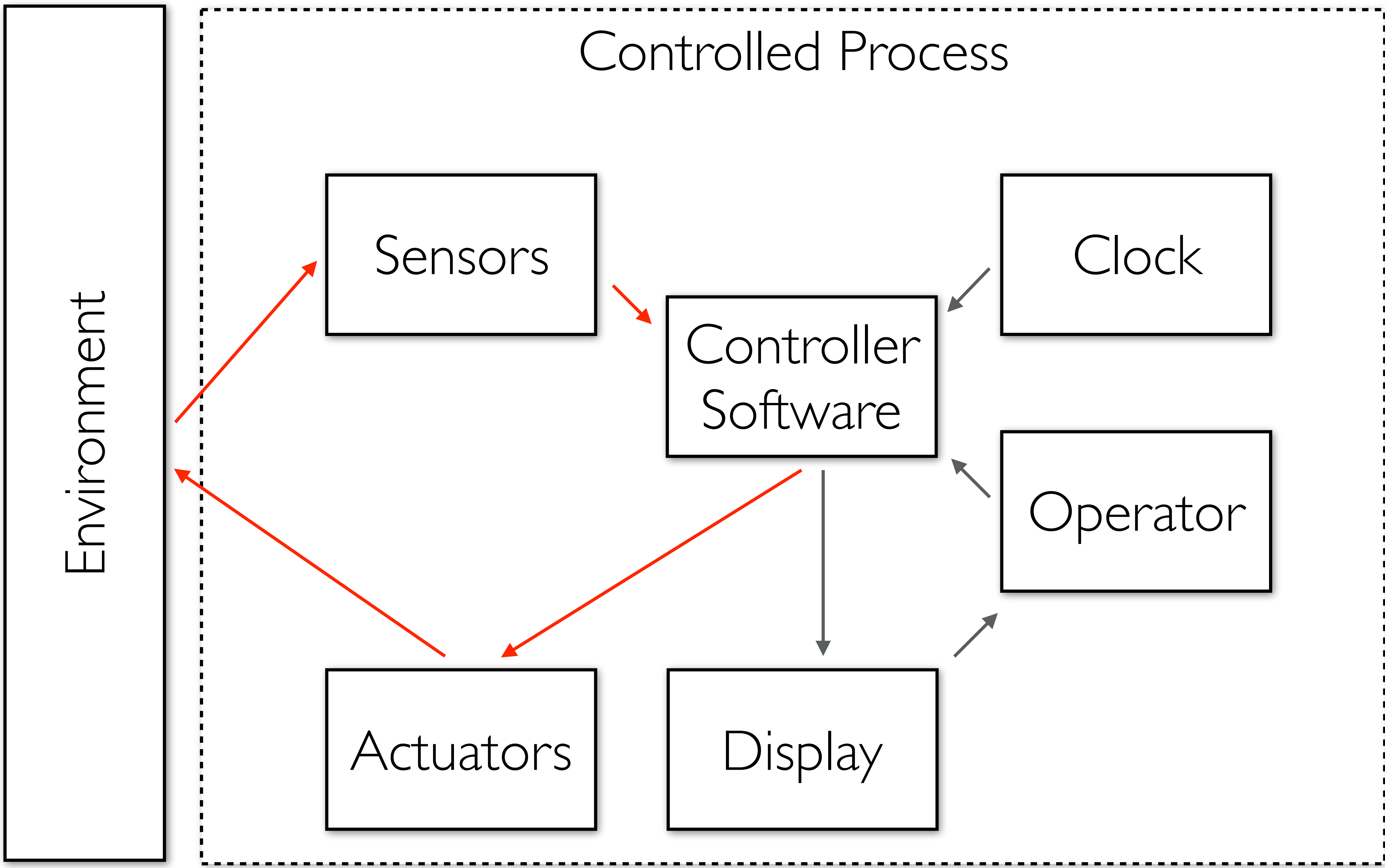
CONTROL THEORY

DESIGN OF SOFTWARE FOR EMBEDDED SYSTEMS (SWES)

Dr. Peter Tröger
Operating Systems Group, TU Chemnitz

EMBEDDED SYSTEM

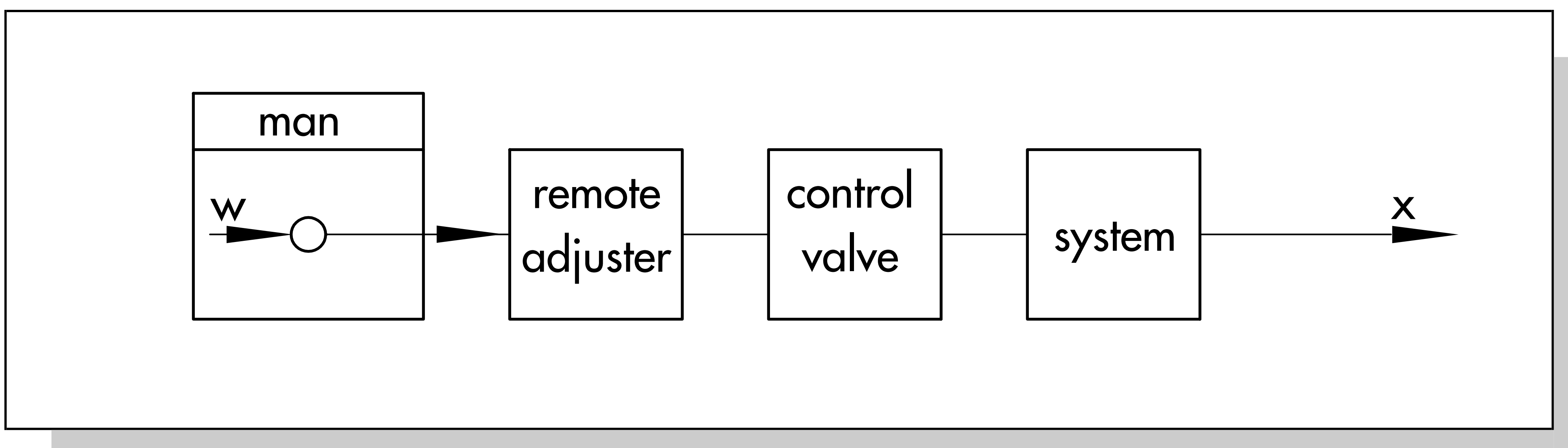


CONTROL THEORY

- Create abstract models for a controlled process
  - Typical challenge in ‚connected' embedded systems
  - Focus on static and dynamic behavior
  - Results in the behavioral design of a controller
- Abstraction from specific implementation details (e.g. car speed control) creates a dedicated mathematical problem
  - How to consider measured input to control some regulation output
  - Mathematical principles as glue between engineering disciplines
  - Everybody has some kind of ‚control problem'
  - Example: Control unit (Steuergerät) in automotive systems

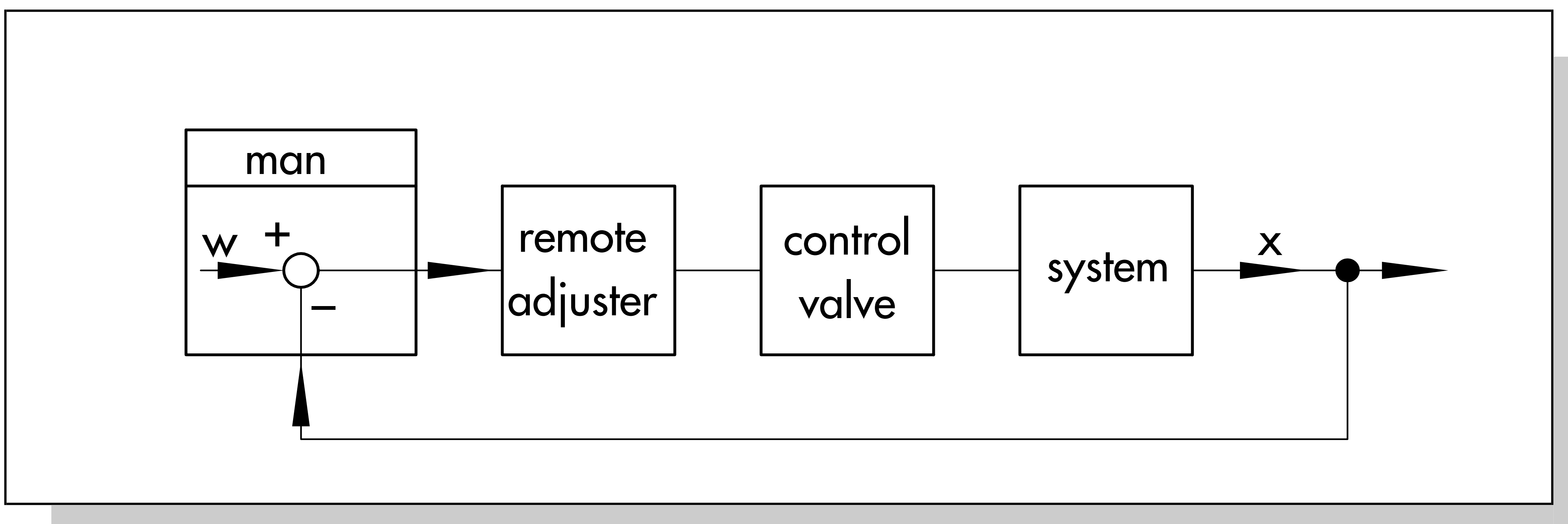
CONTROL THEORY

- **Open loop / non-feedback control („Steuerung“)**
  - Controller activity based on current state
  - Output of the controlled system is not observed
  - External deviations must be considered at design-time
  - Example: Power supply for electric motor with constant load

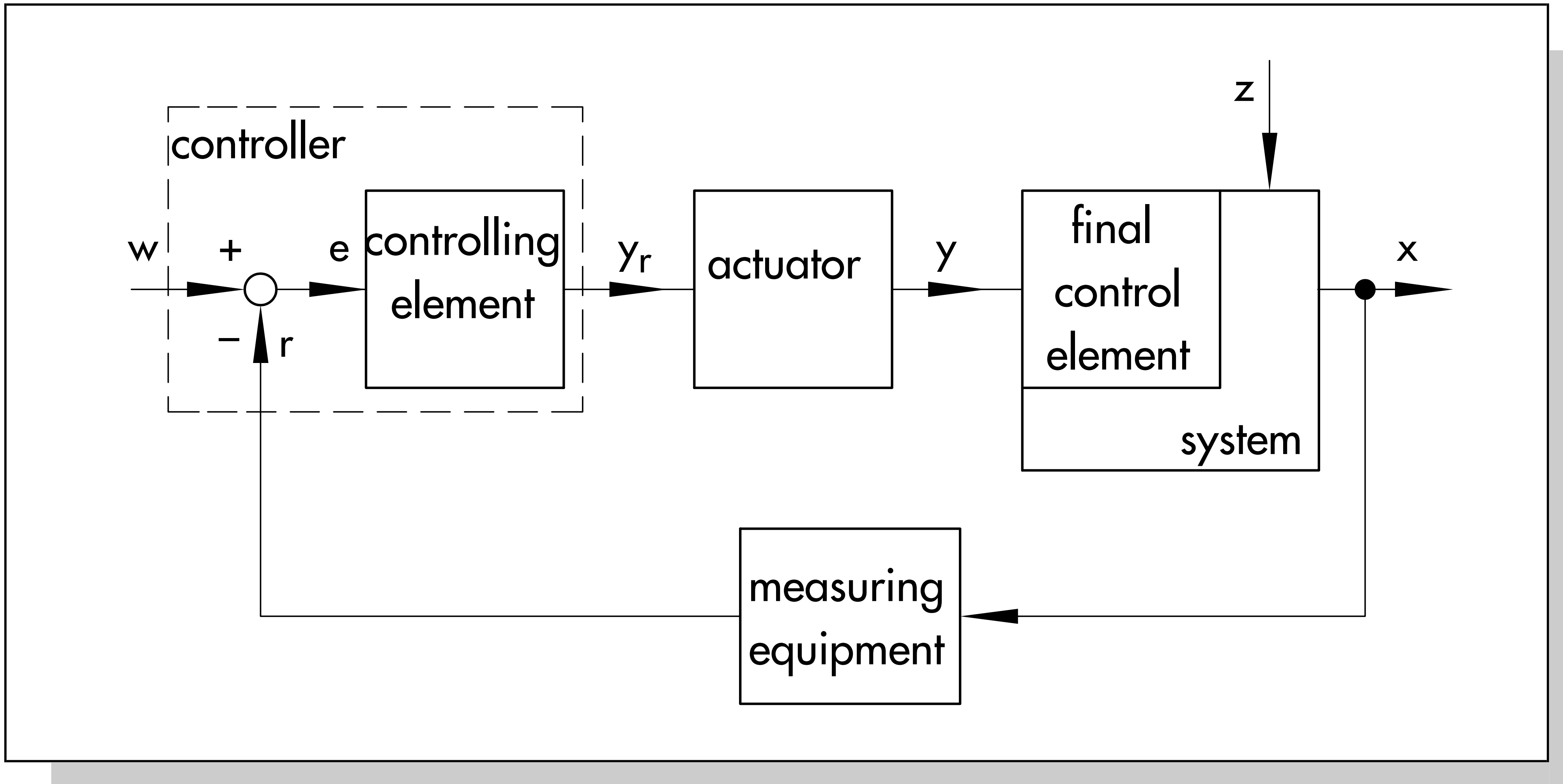


CONTROL THEORY

- **Closed loop / feedback control („Regelung“)**
  - Feedback from system output used to adjust controller operation
  - Error between output and reference used to adopt the operation
  - System can react on external disturbances
  - Instability can happen



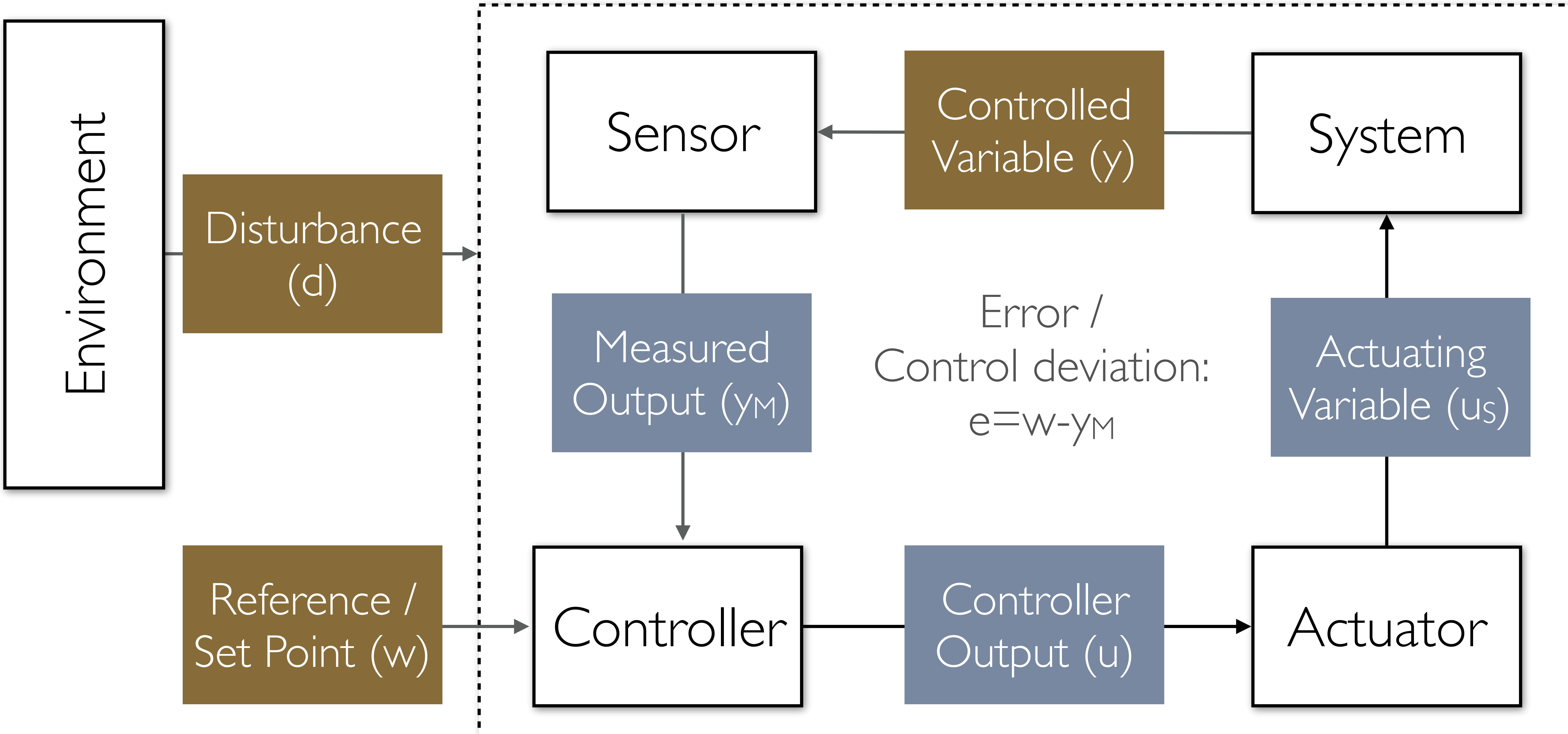
CLOSED LOOP CONTROL



- Regulatory control: Manage with respect to a reference value (w)
- Disturbance rejection: Eliminate effect of a disturbance (z)
- Optimization: Achieve the „best“ value of the outputs (x)

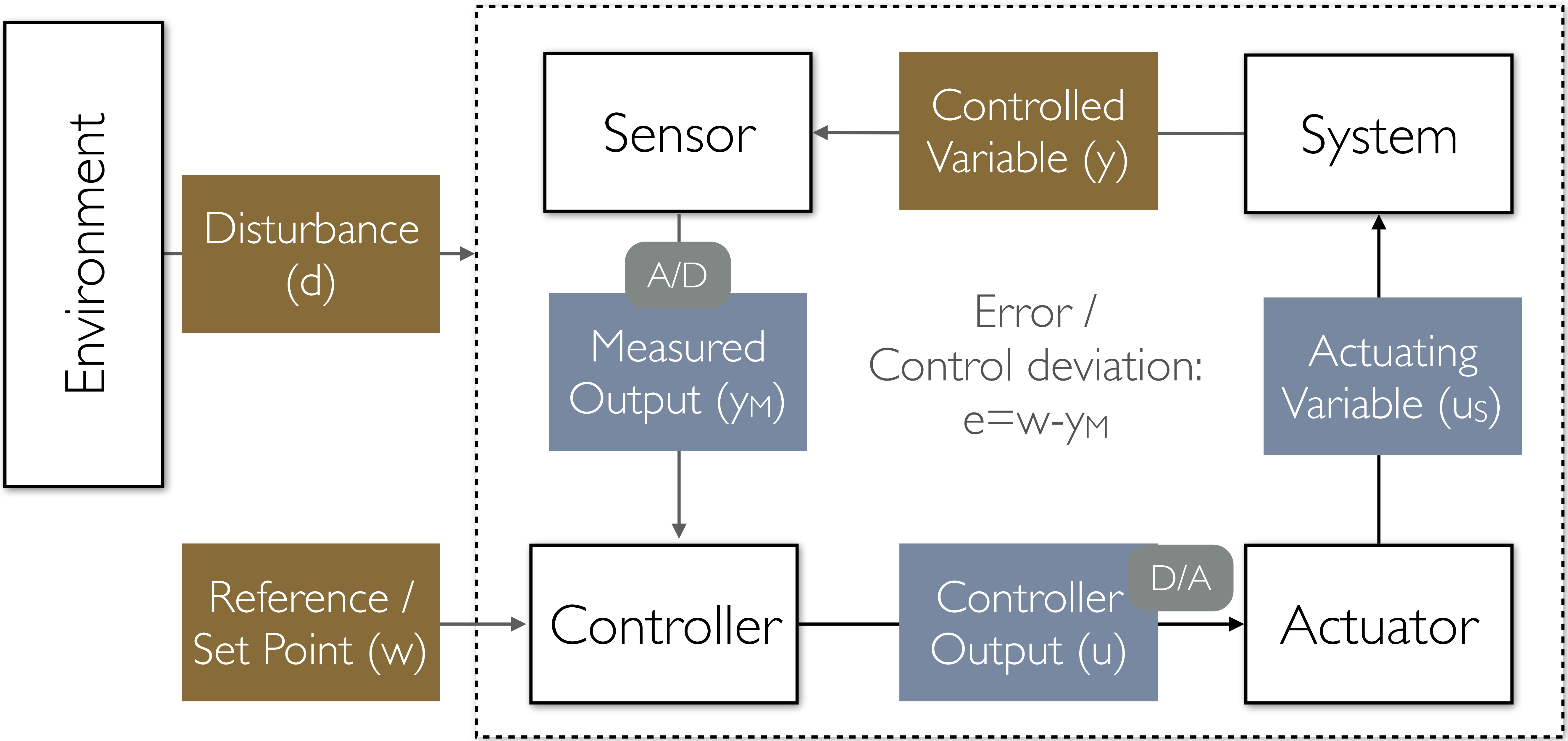
[Samson AG]

CLOSED LOOP CONTROL



Symbols differ in German DIN notation and English notations!

DIGITAL CONTROLLER

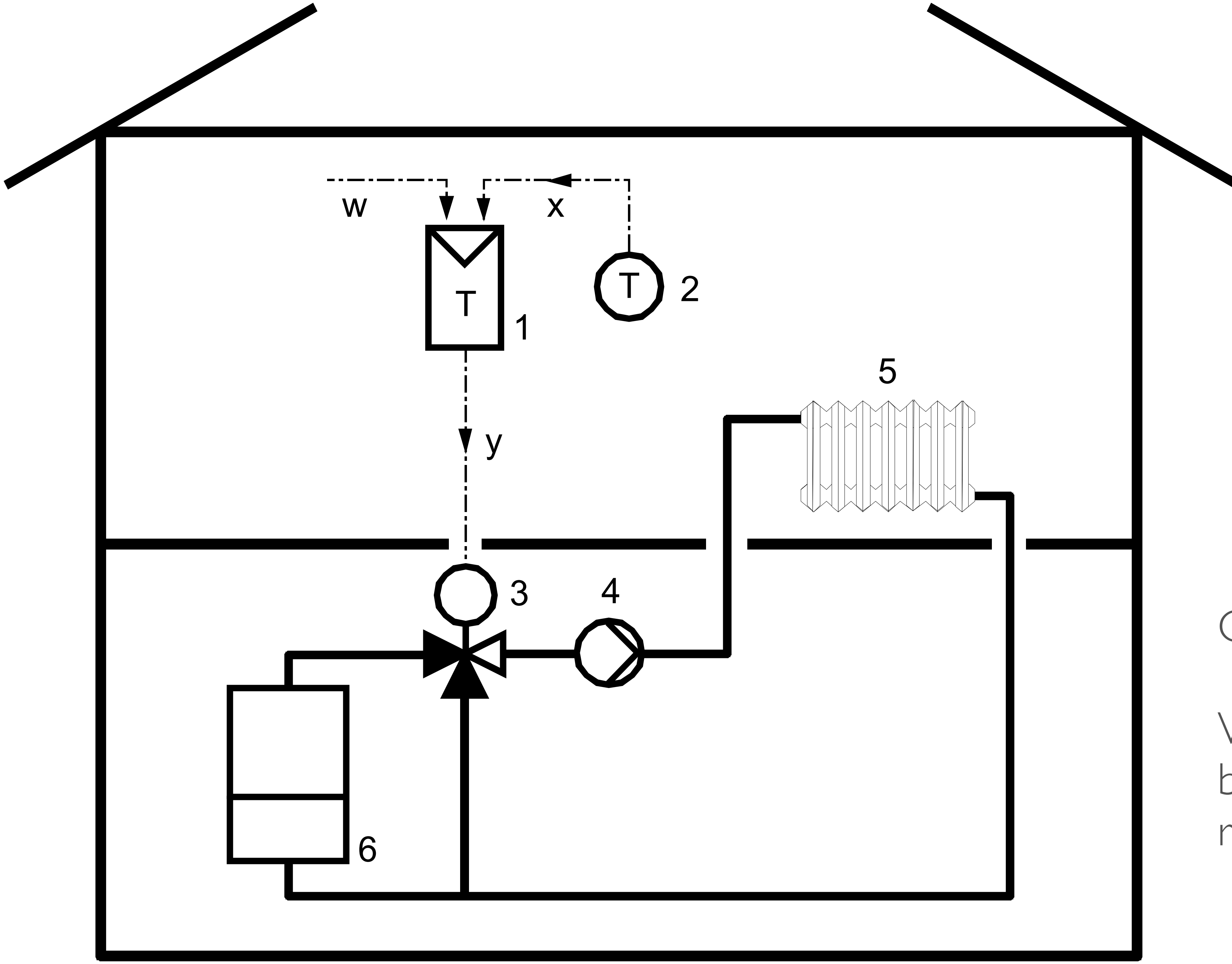


Symbols differ in German DIN notation and English notations!

CLOSED LOOP CONTROL

- **Set point** given by user, or higher-level system, or both
- Examples
  - Biology: Upright walk, body temperature, eye adaption on light
  - Home devices: Heating based on internal temperature sensor, fridges
  - Industry: ABS in cars, power grid control, servo systems
- Time dependency of set point
  - Constant set point: **Fixed set point control**
    - Example: Boiler temperature control
  - Varying set point: **Follow-up control**
    - Example: Weather-compensated temperature control

EXAMPLE



- 1 Temperature controller
- 2 Temperature sensor
- 3 Mixing valve
- 4 Circulation pump
- 5 Radiator
- 6 Boiler

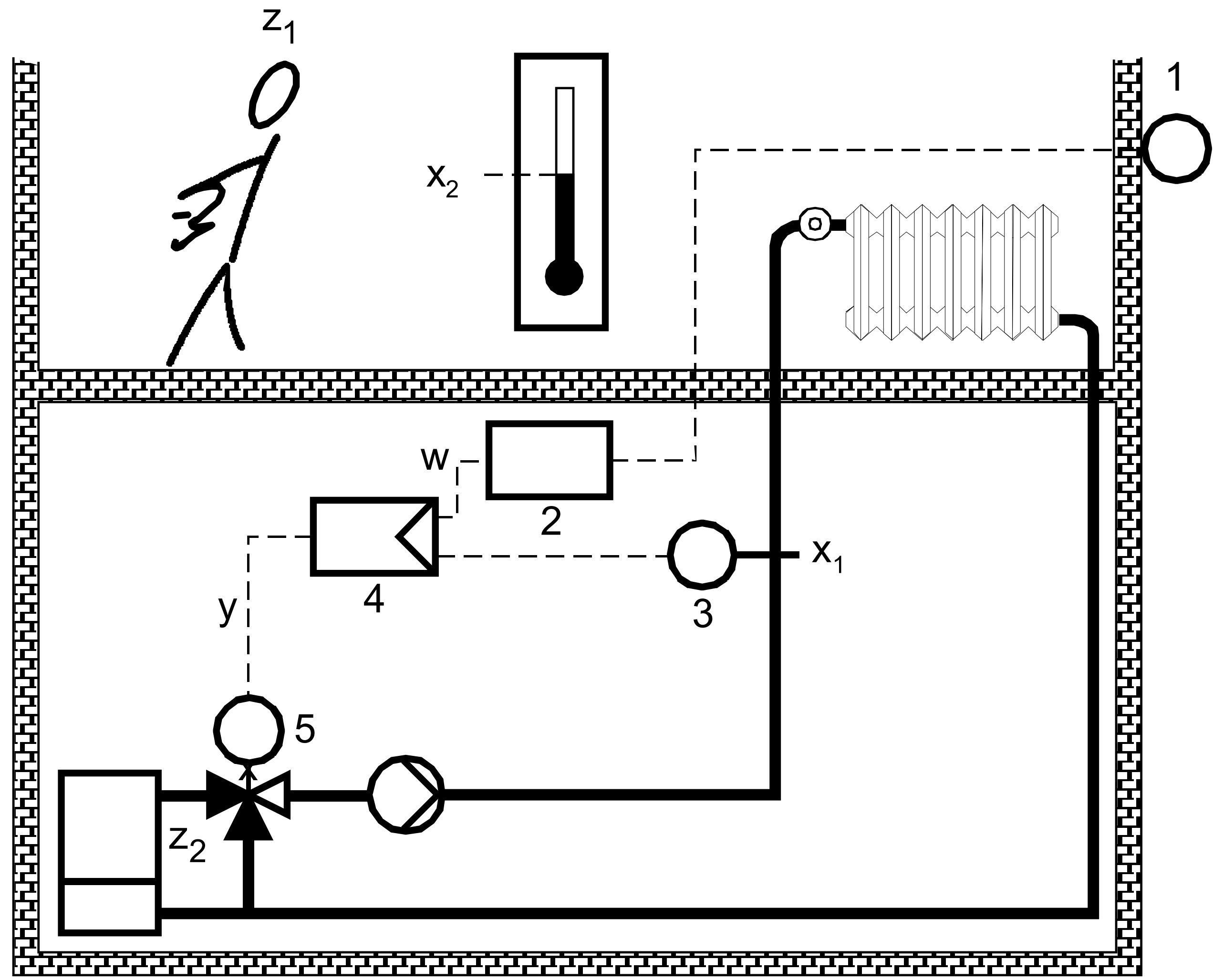
Closed loop clearly identifiable

Valve is opened and closed to bring controlled variable and reference variable together

[Siemens]

EXAMPLE

- 1 - Outside temperature sensor
- 2 - Heating curve setting
- 3 - Supply temperature sensor
- 4 - Supply temperature controller
- 5 - Control valve (actuator + control element)
- w - Supply temperature setpoint (open-loop control variable 1)
- x_1 - Supply temperature actual value (closed-loop controlled variable)
- x_2 - Room temperature actual value (open-loop control variable 2)
- z_1 - Interference variable 1 (external heat gain)
- z_2 - Interference variable 2 (fluctuating boiler water temperature)
- y - Manipulated variable



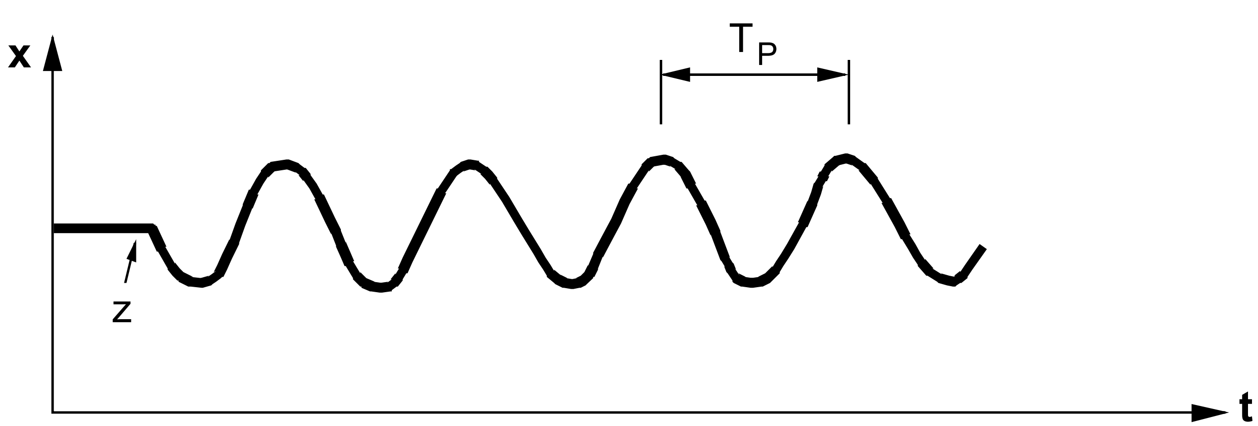
[Siemens]

VARIATIONS

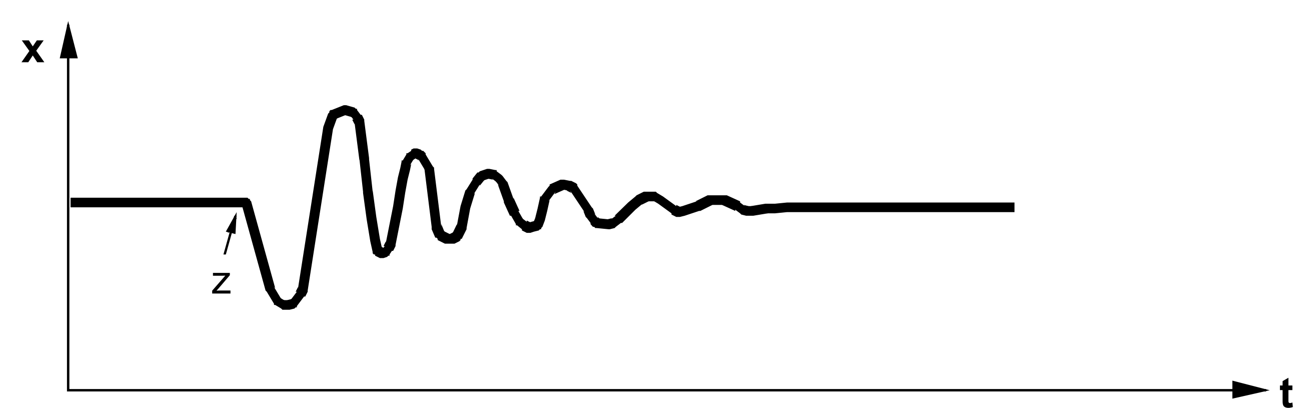
- **Disturbance feed-forward control**
  - Disturbance is directly used by controller, allows faster reaction
  - Disturbance must be **locatable** and **measurable**
- **Cascade control**
  - One controller feeding another one
- **Robust control, adaptive control**
  - Modification of controller through self-feedback
  - Example: Flying missile with decreasing fuel mass

STABILITY

- Control loop is an oscillatory system
  - Controlled and manipulated variable influence each other
  - System therefore may escalate into oscillation -> **unstable system**
- **Bounded input / bounded output (BIBO) stability**
  - Output signal should not grow indefinitely when the input is limited
  - Mandatory condition for every serious control system



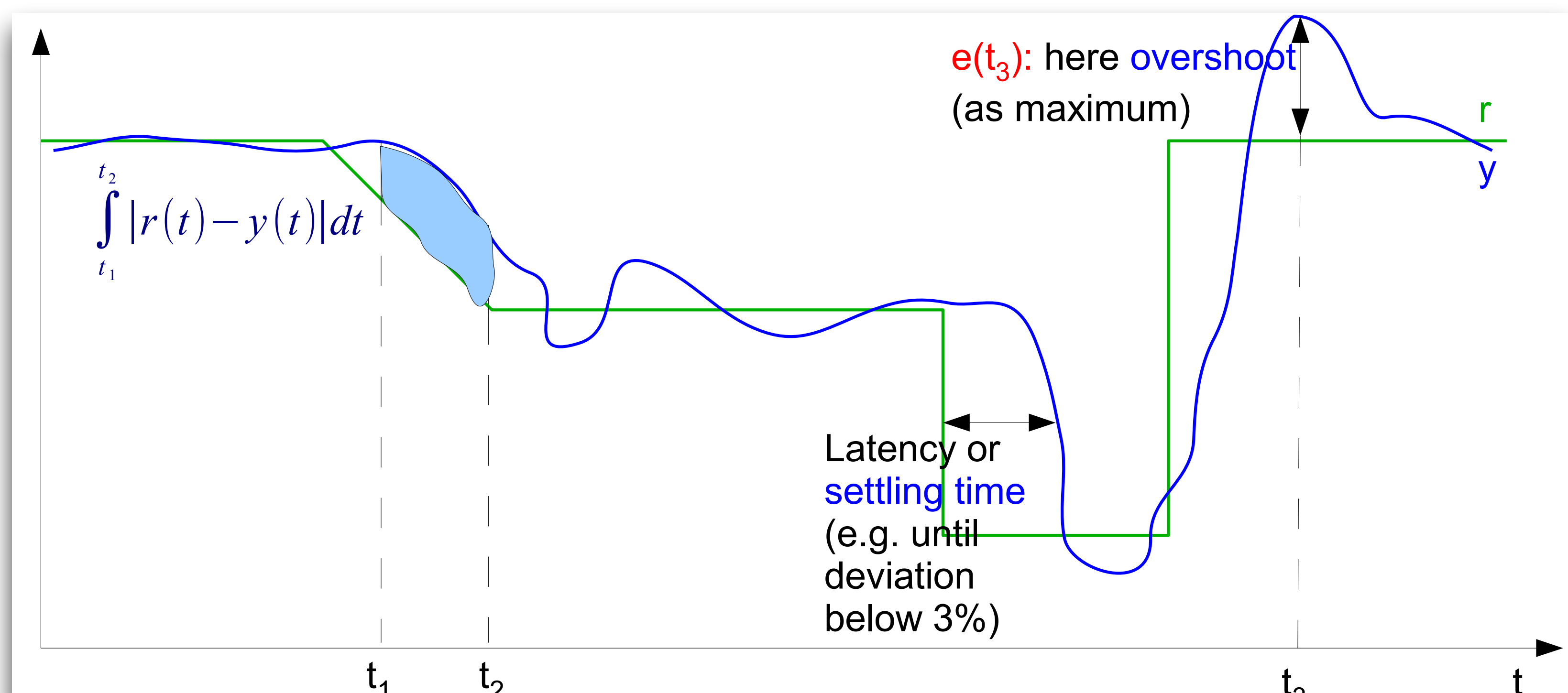
Undamped oscillation



Damped oscillation

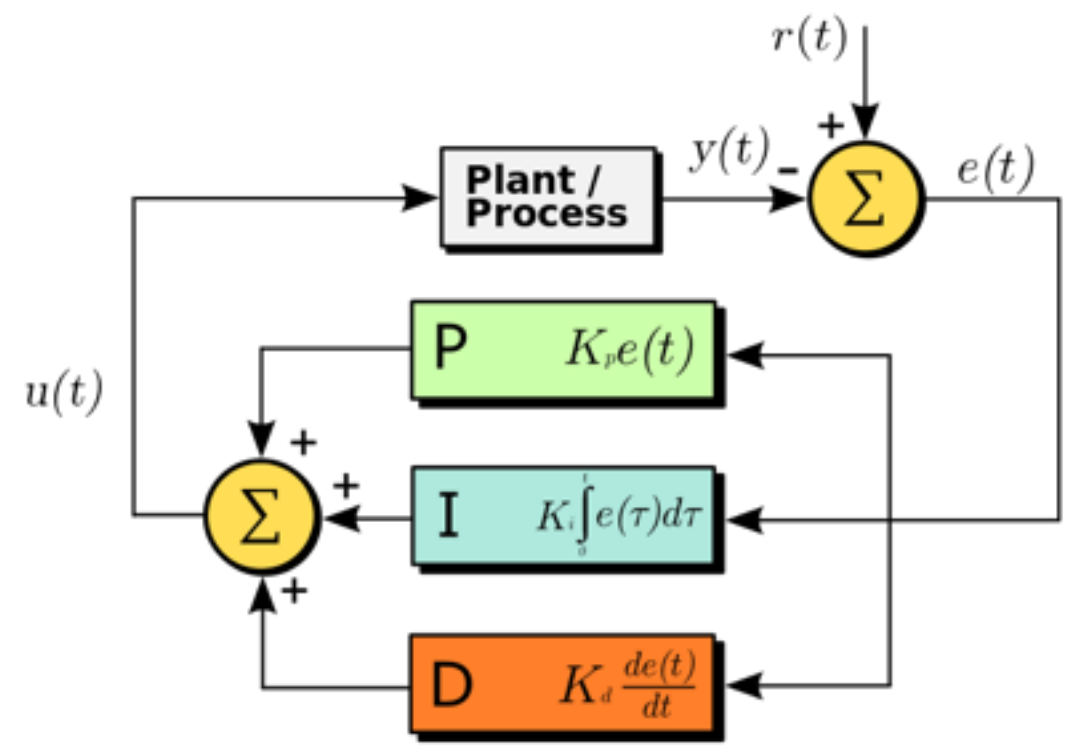
CONTROL QUALITY

- Places of disturbance may vary
  - Influence on measured output, system input or system output
- Control deviation should be minimized over time
  - Especially relevant for real-time systems



CONTROLLER

- How to create a reasonable controller ?
  - Desired value of **controlled variable** given by **set point**
  - Compute new value of a **control variable** to correct the **error** between **measured variable** and set point
  - Often, controlled variable is physical, while measured variable is the representation of it
- Create a **control path** from basic elements
  - **Proportional (P) element**
  - **Integral (I) element**
  - **Derivative (D) element**



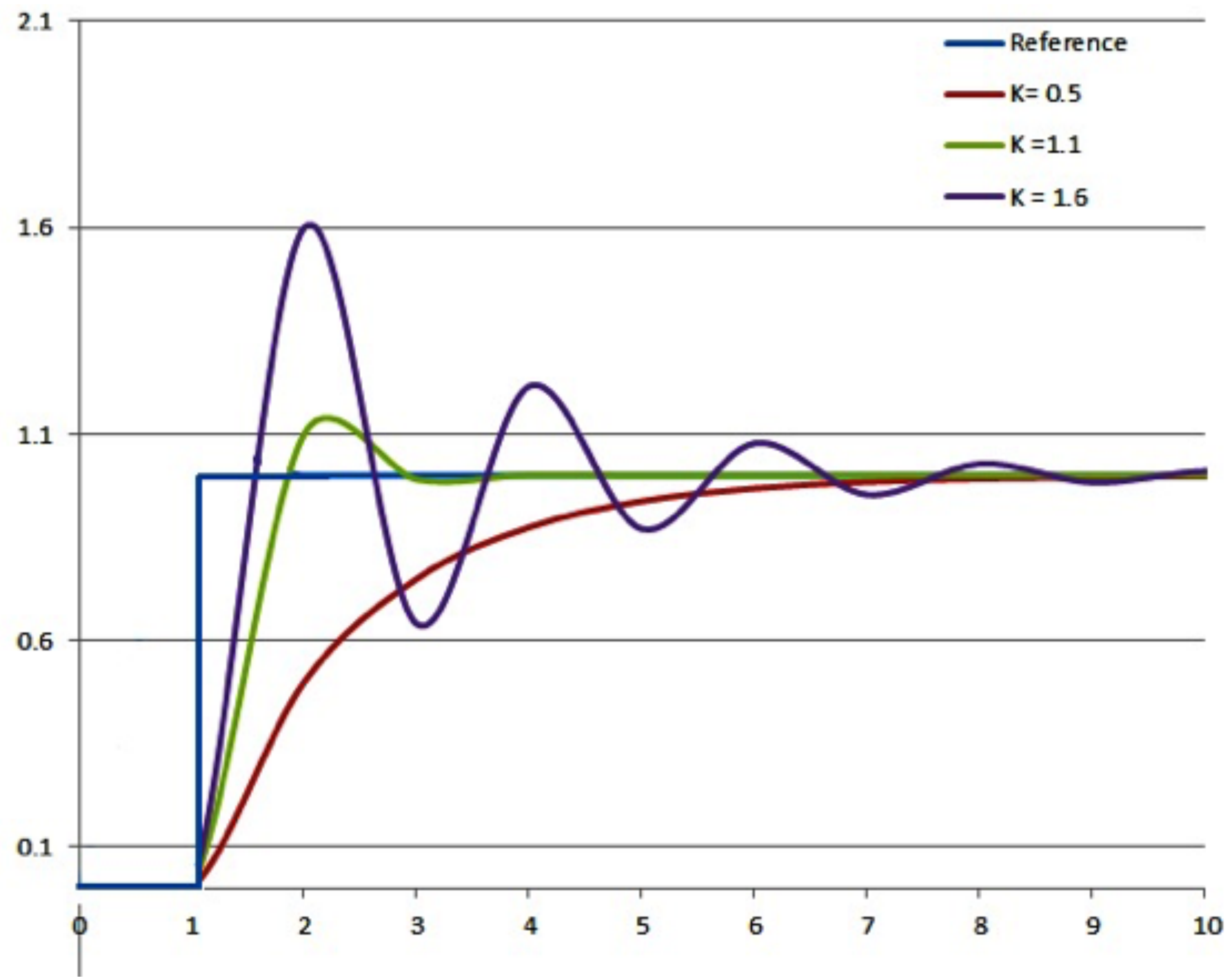
[Wiikipedia]

ELEMENTS

- **Proportional (P) element**
  - Perform immediate proportional reaction to error at output
  - Adjusted by **gain factor k_P**
- **Integral (I) element**
  - Accumulate past errors and react on them
  - Adjusted by **gain factor k_I**
- **Derivative (D) element**
  - Based on current rate of change of the error, predict the future
  - Adjusted by **gain factor k_D**

PROPORTIONAL ELEMENT

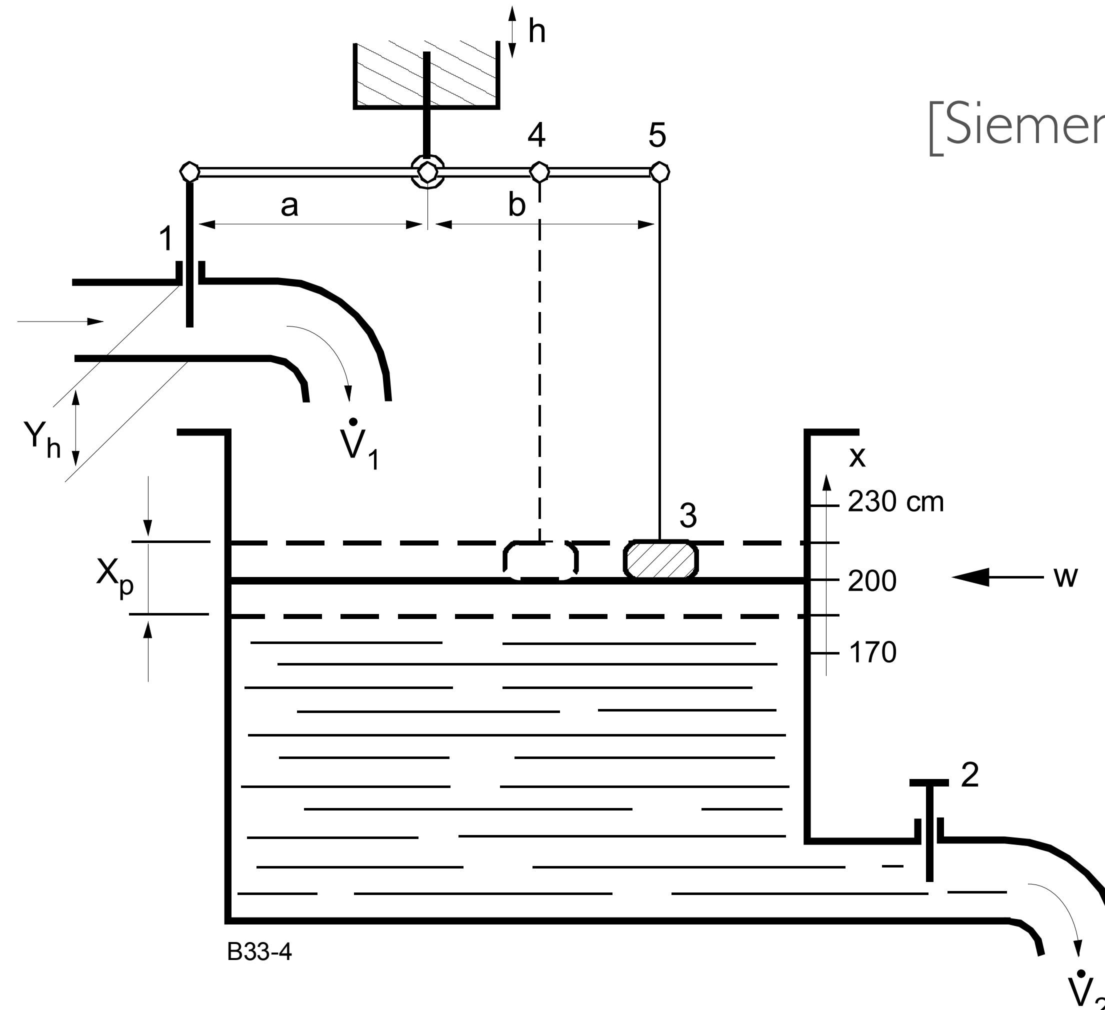
- Output is proportional to the current error
- Gain factor too low: Reaction not good enough to fix the error
- Gain factor too high: Reaction to error is too extreme (**stability**)
- Only instantaneous reaction, therefore **steady-state error**



$$P_{\text{out}} = K_p\, e(t)$$

P CONTROLLER

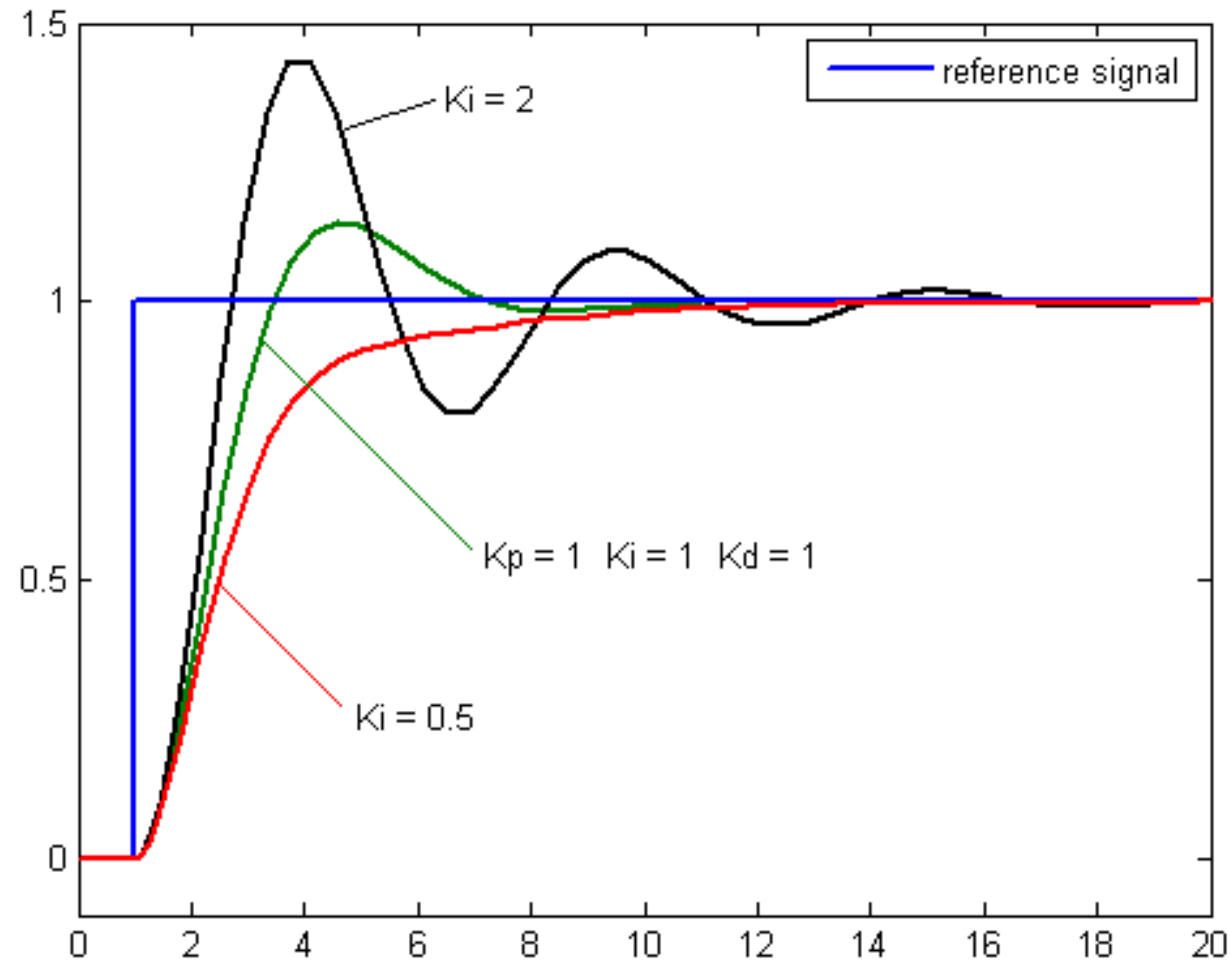
- Example: Mechanical water level control systems
  - Adjustable inflow with valve 1
  - Variable outflow with valve 2
  - Float sensor 3
  - Gate valve as control element
  - Controlled variable is water level x
  - Output variable y is position of gate valve
  - Keep water level constant, regardless of water drawing
  - Desired water level (set point) defined by lever attachment height h



[Siemens]

INTEGRAL ELEMENT

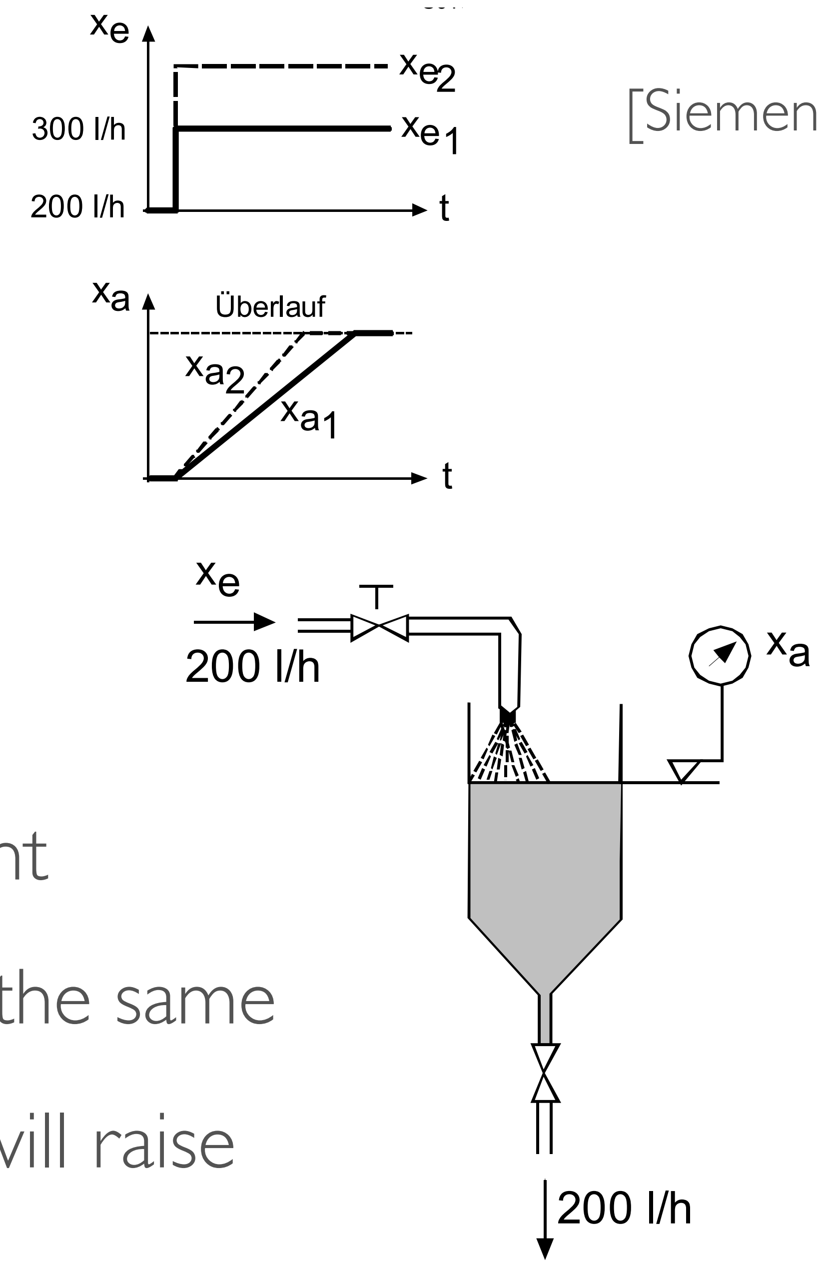
- Output is proportional to past error behavior
- Delayed reaction to accumulated errors
- Can fix steady-state error resulting from proportional reaction
- May produce output that exceeds the target (**overshoot**)



$$I_{\text{out}} = K_i \int_0^t e(\tau)\, d\tau$$

I CONTROLLER

- Integral-action controller
  - Output variable formed from sum of consecutive input variables over time
- Example: Tank with inlet and outlet
  - Flow rate at inlet as input variable
  - Level at output as output variable
  - With constant flow rate, level remains constant
  - With increased incoming flow, output flow is the same
  - With input greater than discharge rate, level will raise
    - Proportional reaction no longer suitable
    - Needs integral reaction



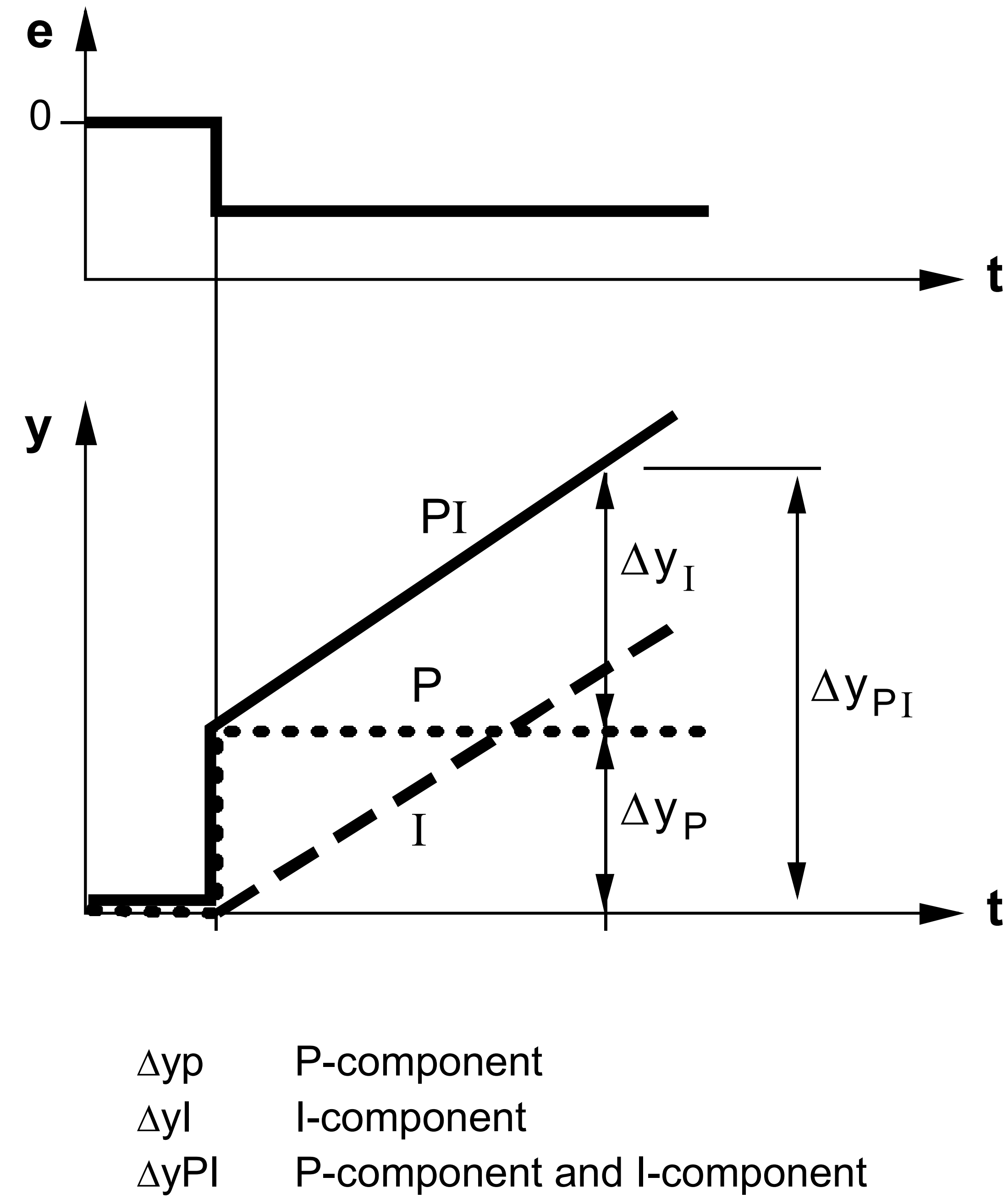
[Siemens]

CONTROLLERS

- P controllers are **load-dependent**
  - Controlled variable cannot be kept constant at all loads
  - Output variable is proportional to the input variable
  - Quick reaction
  - Residual derivation: **Steady-state deviation**
- I controllers are **load-independent**
  - Control action builds up slowly, therefore **time-dependent**
  - Manipulated variable changes as long as deviation is given

PI CONTROLLER

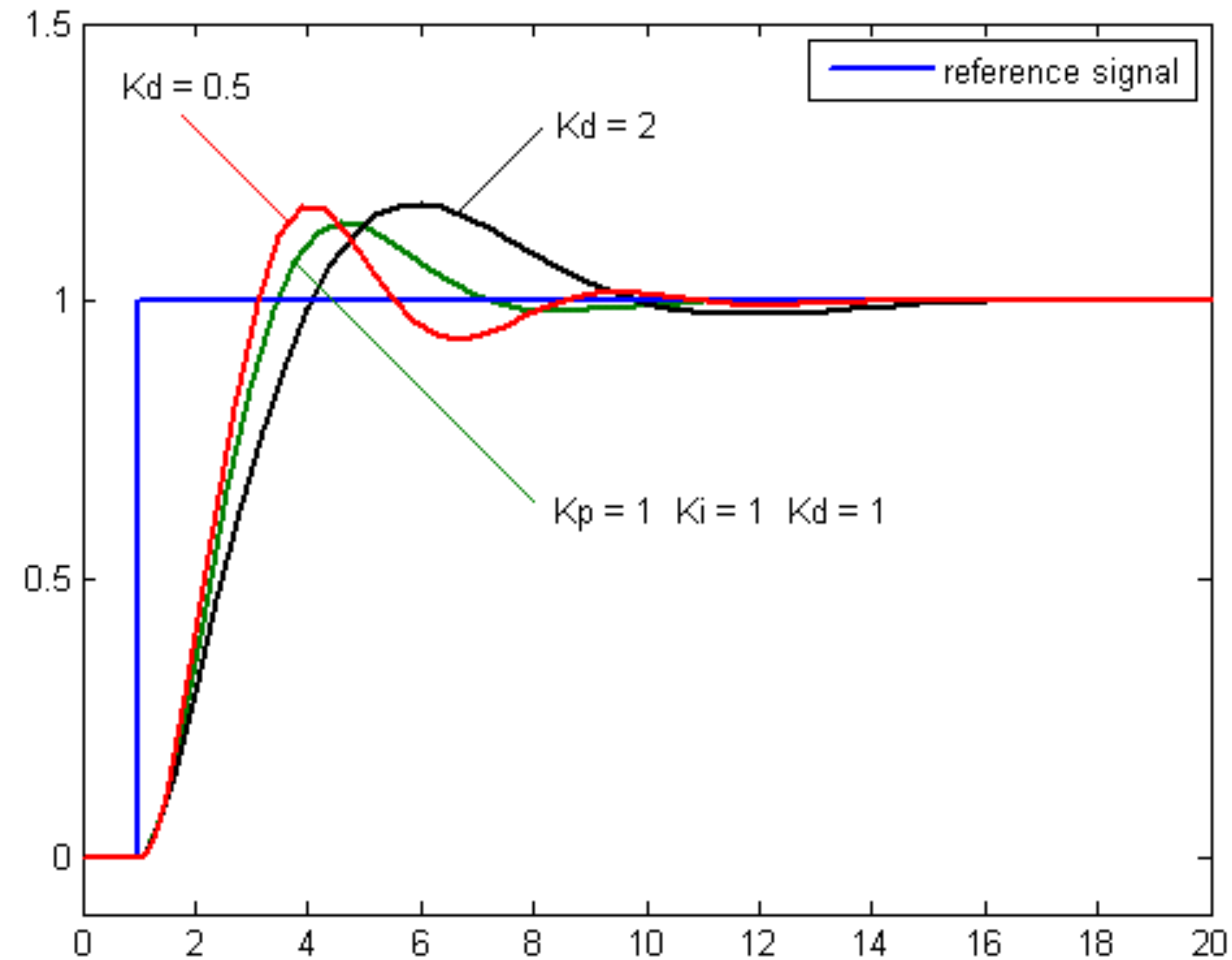
- Combine best of both worlds: **PI controller**
  - Remove steady-state error, get medium speed of reaction
  - Fast P controller + load-independent I controller
  - Can be implemented in different ways
    - Mechanical / hydraulic / electrical / electronic solution



[Siemens]

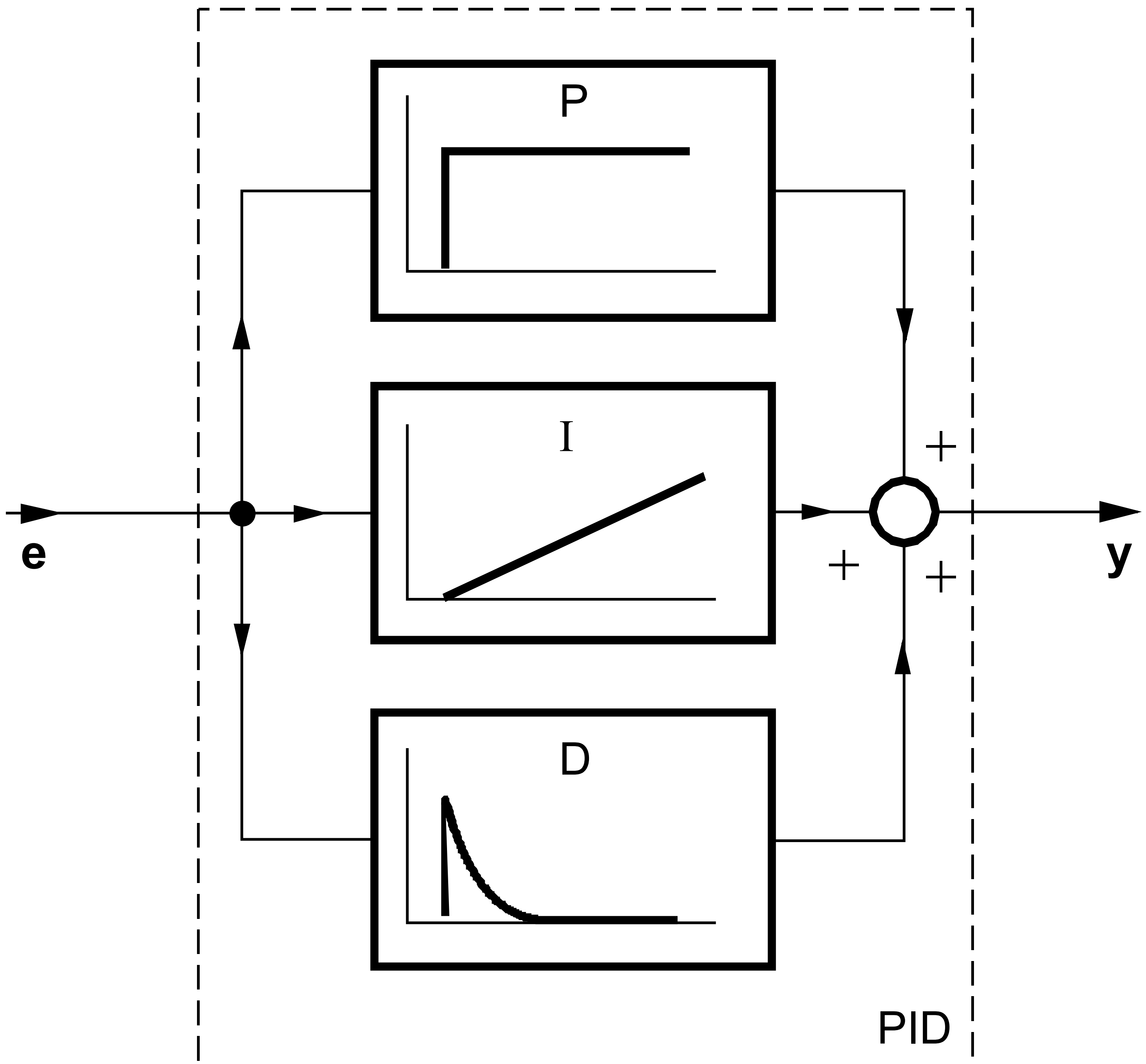
DERIVATIVE ELEMENT

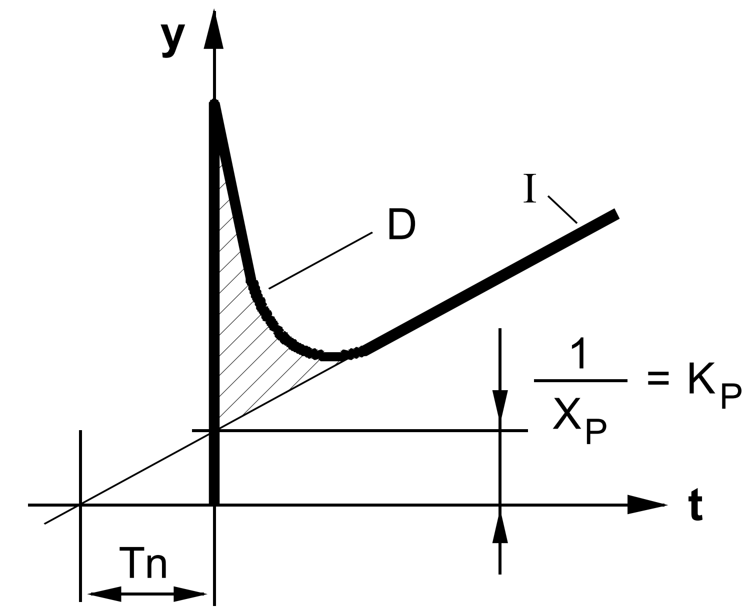
- Compute derivative („slope of change“) of the error
- Again adjusted by gain factor
- Initially, the error changes dramatically, so D-element compensates directly
- Afterwards, influence reduces and P+I parts are ‚taking over‘



$$D_{\text{out}} = K_d \frac{d}{dt} e(t)$$

PID CONTROLLER





- Initial charge by D
- Reduction almost to P level
- Rises again from I influence

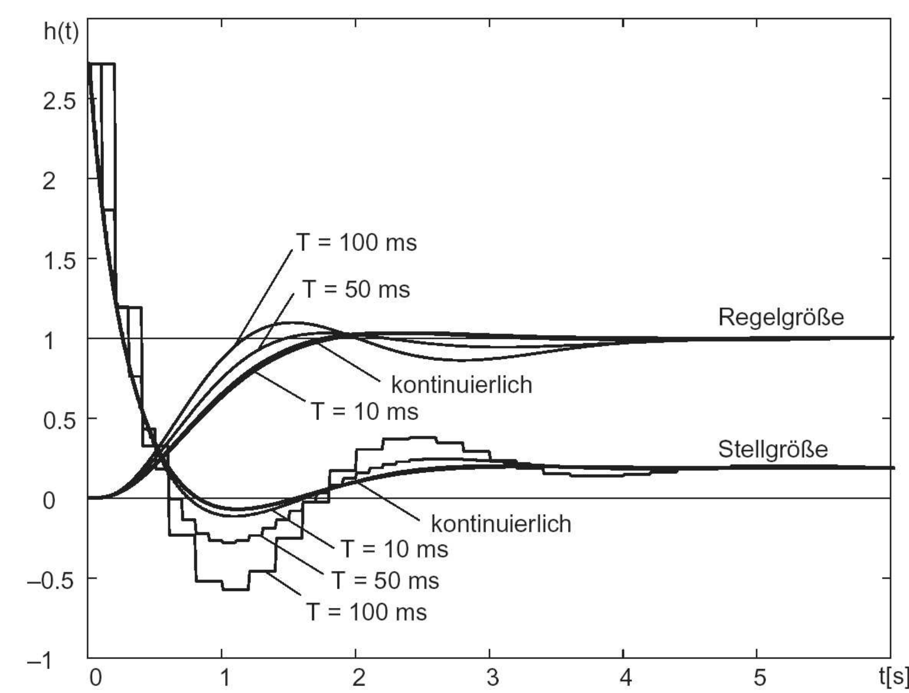
[Siemens]

PID CONTROLLER

- Date back to 1890 governor design and ship steering
- Based on observation of human ship controller
  - Compensates based on current error, past error and change rate
- Optimum behavior
  - **Disturbance rejection** - Stay at a given setpoint
  - **Command tracking** - Implement setpoint changes
    - **Rise time** - How fast going close to the final value
    - **Settling time** -
      How fast settling into some range around the final value

DIGITAL PID CONTROLLER

- Digital microcontrollers implement control function
  - Easier realization of non-linear behavior and adaptive control
  - Reconfiguration of software possible
  - Time-discrete, quantized behavior
- A/D and D/A conversion requires sampling
  - Danger of instability through phase shift
  - Restricted data word length
  - Accumulation of rounding errors
  - Quantization errors
  - Overflow in calculation with wrap-around



ADJUSTING THE CONTROLLER

- Controller adjustment by determining gain factors / coefficients
- Heuristic method by Ziegler / Nichols for P / PI / PID controllers
  - Only suitable for stable systems, focus on disturbance compensation
  - I-gain and D-gain set to zero, increase P-gain until oscillation
  - Table lookup for gain parameters, based on oscillation period
- Empirical method
  - Response too slow:-> Increase influence of P component, reduce influence of I component afterwards
  - Response oscillates slowly towards goal signal -> Increase influence of P component, reduce influence of D component afterwards
  - Overshooting -> Reduce influence of P component, increase influence of I component afterwards