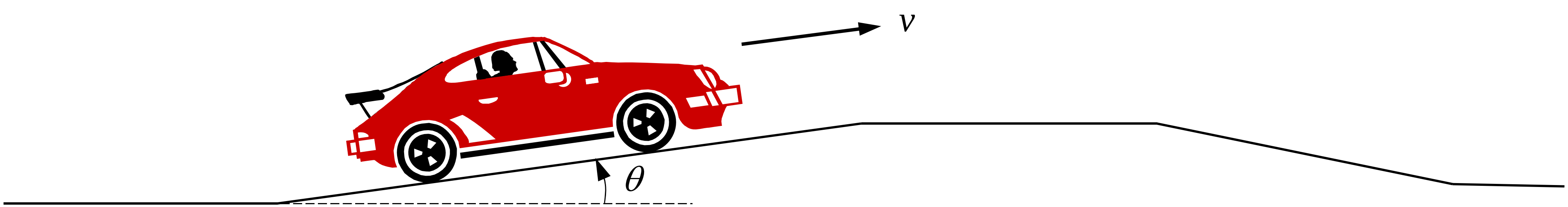
Chapter 2

Our first control design

Design problem 1: cruise control (1)



Process input:	gas pedal (throttle)	u
Process output:	cruising velocity	v
Desired output (reference signal):	desired velocity	v_r (constant)
Disturbance:	slope	θ

What do we expect from a closed-loop cruise control?

Think for a minute ...

Design problem 1: cruise control (2)

What do we expect from a closed-loop cruise control?

- Reduce effects of process disturbances (slope)
- Make system insensitive to process parameter variations (e.g. load weight)
- Create well-defined dynamic relations between output and reference ("smoothed" transient responses)
- But: Risk for instability due to feedback!

To do:

- Draw block diagram of the uncontrolled and controlled system
- Analyze the open and closed-loop system behavior
- Use a PID control (Proportional-Integral-Differential control)

$$u = k_P(v_r - v) + k_I \int_0^t (v_r - v)d\tau + k_d \frac{\mathrm{d}e}{\mathrm{d}t}$$

- Determine proper controller parameters
- Check the realizability!

LRS
Prof. Dr.-Ing. Steven Liu
Lehrstuhl für Regelungssysteme
Technische Universität Kaiserslautern

The amazing properties of PID control

Proportional action:

- compensates immediately for the current control error (present effect)

Integral action:

- accumulated effect, eliminates the steady-state control error, i. e., for $u(t) = u_0 = \text{const.}$ and $e(t) = e_0 = \text{const.}$ we must have: $e_0 = 0$!

Can you explain this amazing property of integral action?

...

Differential action:

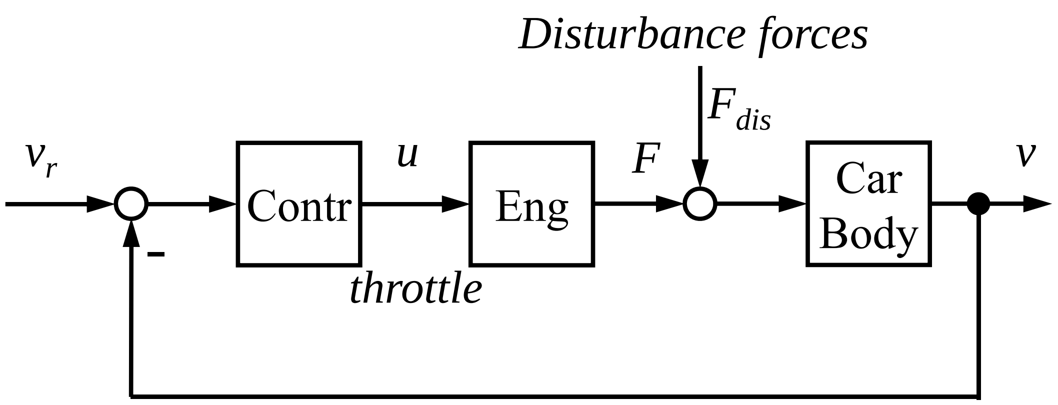
- compensates for future control error (prediction by linear extrapolation)

PID controller is simple and very widely used. For complex systems and demanding requirements more sophisticated controllers can be applied.

Design problem 1: cruise control (3)

Construction of a block diagram:

- Understand how the system works
- Identify the major components and relevant signals
- Key questions: what is the essential dynamics?
- what are appropriate abstractions?
- Describe the dynamics of the blocks



Simplification assumptions:

- Essential dynamics relates velocity and force
- The force responds very fast to a change in the throttle
- All relations are assumed to be linear (small signal behavior)

Design problem 1: cruise control (4)

Process model:

A simplified mathematical description

$$m\frac{dv}{dt} + Dv = F - mg\sin\theta \approx F - mg\theta$$

With reasonable parameters

$$\frac{dv}{dt} + 0.02v = u - 10\theta$$

where

v [m/s]	speed
$0 \leq u \leq 3$ [m/s^2]	normalized throttle
θ [rad/s]	slope

The closed-loop system behavior:

P (Proportional) controller:

$$u = k_P(v_r - v)$$

The closed-loop system is described by

$$\frac{dv}{dt} + (0.02 + k_P)v = k_P v_r - 10\theta$$

In steady-state with $t \to \infty$ we have

$$v_\infty = \frac{k_P v_r}{0.02 + k_P} - \frac{10\theta}{0.02 + k_P}$$

Active excise

Stop and think!

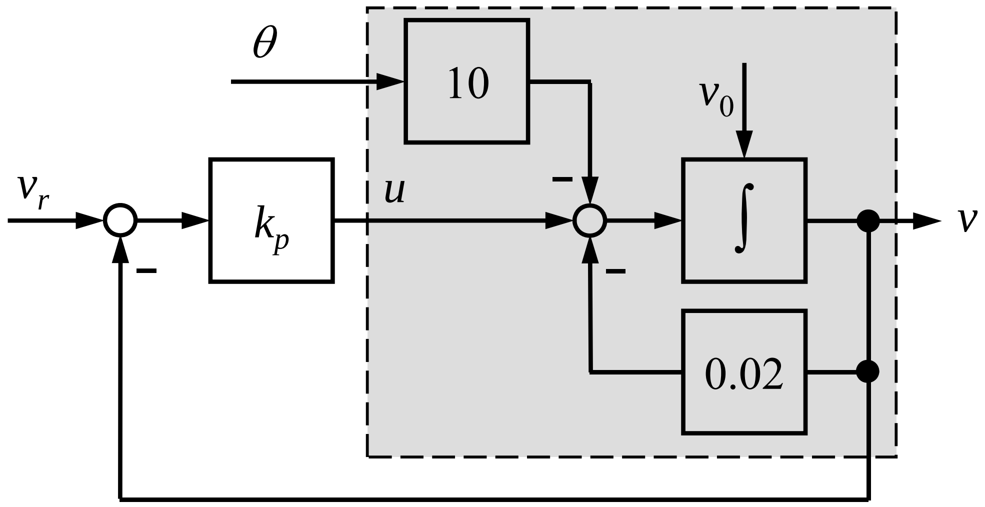
How is the behavior of the equations

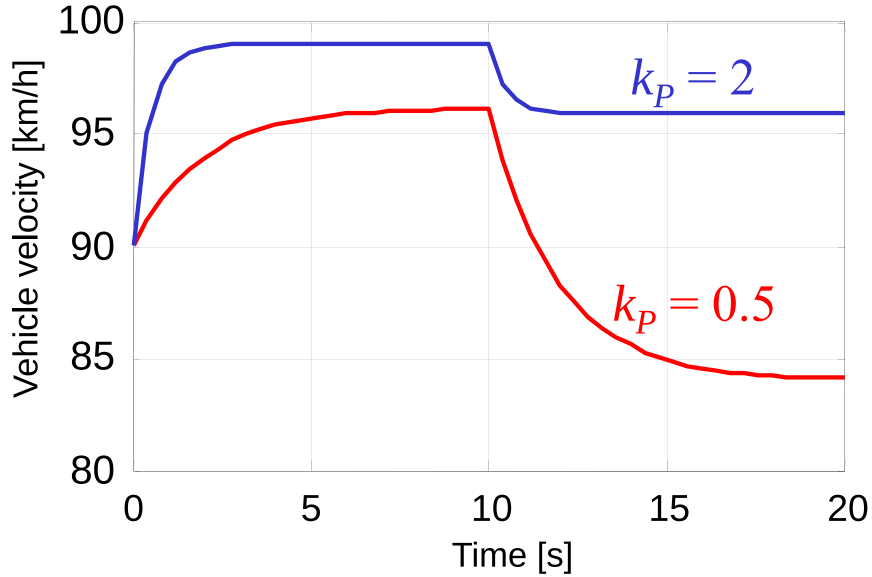
$$\frac{dv}{dt} + 0.02v = u - 10\theta \quad \text{and} \quad \frac{dv}{dt} + (0.02 + k_P)v = k_P v_r - 10\theta$$

and how is the closed-loop behavior influenced by the controller parameter k_P?

How should we choose k_P to achieve a "good" control behavior?

Design problem 1: cruise control (5)





How to choose the controller parameter?

- The larger the controller gain k_P, the better the dynamic behavior
- The larger the controller gain k_P, the smaller the steady-state velocity error
- Question: Can we choose $k_P \to \infty$?
- No, since throttle variation range is limited $\to$ saturation!
- Steady-state velocity error will remain!

Prof. Dr.-Ing. Steven Liu
Lehrstuhl für Regelungssysteme
Technische Universität Kaiserslautern

Design problem 1: cruise control (6)

Choose a different controller:

PI (Proportional-Integral) controller:

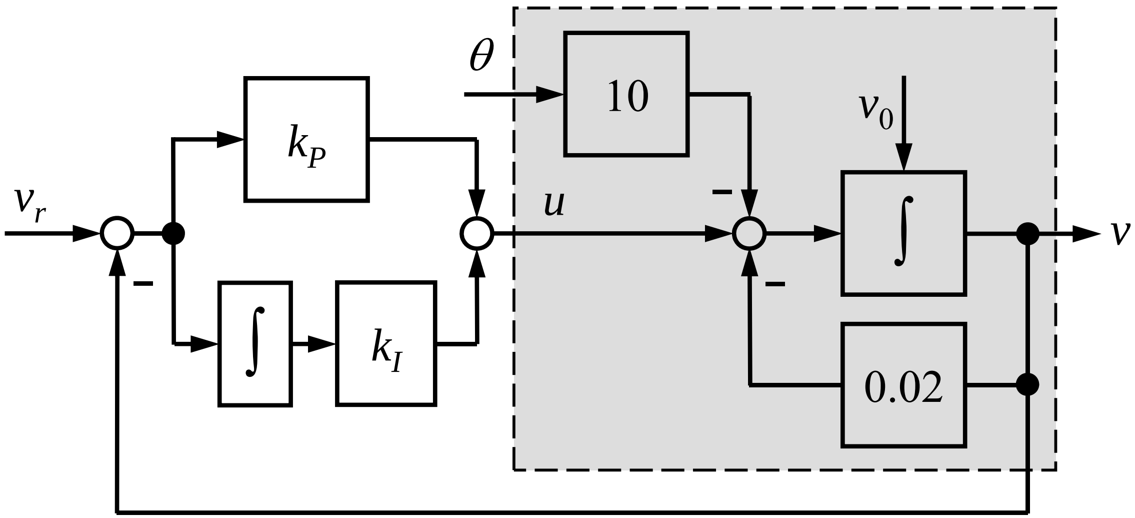
$$u = k_P(v_r - v) + k_I \int_0^t (v_r - v)d\tau$$

The closed-loop system is now described by

$$\frac{d^2v}{dt^2} + (0.02 + k_P)\frac{dv}{dt} + k_I v = k_I v_r - 10\frac{d\theta}{dt}$$

In steady-state with $\theta = \text{const.}$ we have

$$v_\infty = v_r \quad \text{(independent of } v_r \text{ and } \theta\text{!)}$$



How can we determine the performance of the closed-loop control?

The control error can be described by the 2nd order differential equation

$$\ddot{e} + (0.02 + k_P)\dot{e} + k_I e = 10\dot{\theta}$$

Comparison with the normalized resonance oscillator (mass-spring-damper or LCR-oscillator) equation

$$\ddot{x} + 2d\omega_0\dot{x} + \omega_0^2 x = 0$$

gives

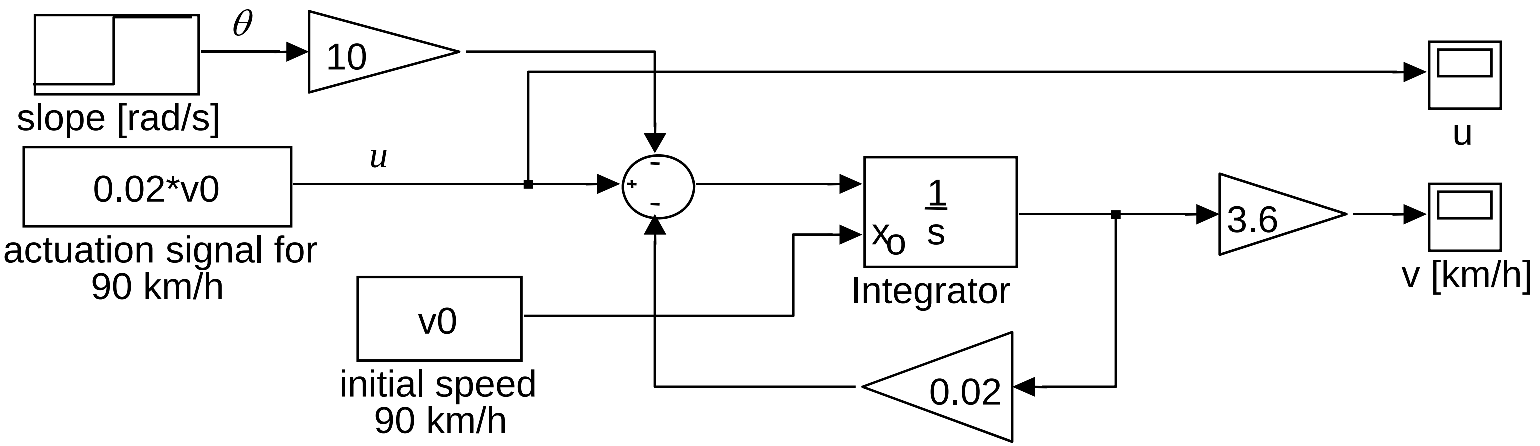
$$k_P = 2d\omega_0 - 0.02 \quad \text{and} \quad k_I = \omega_0^2$$

It is reasonable to choose $0.5 < d \leq 1$ (good damping) and ω_0 as high as possible (fast response).

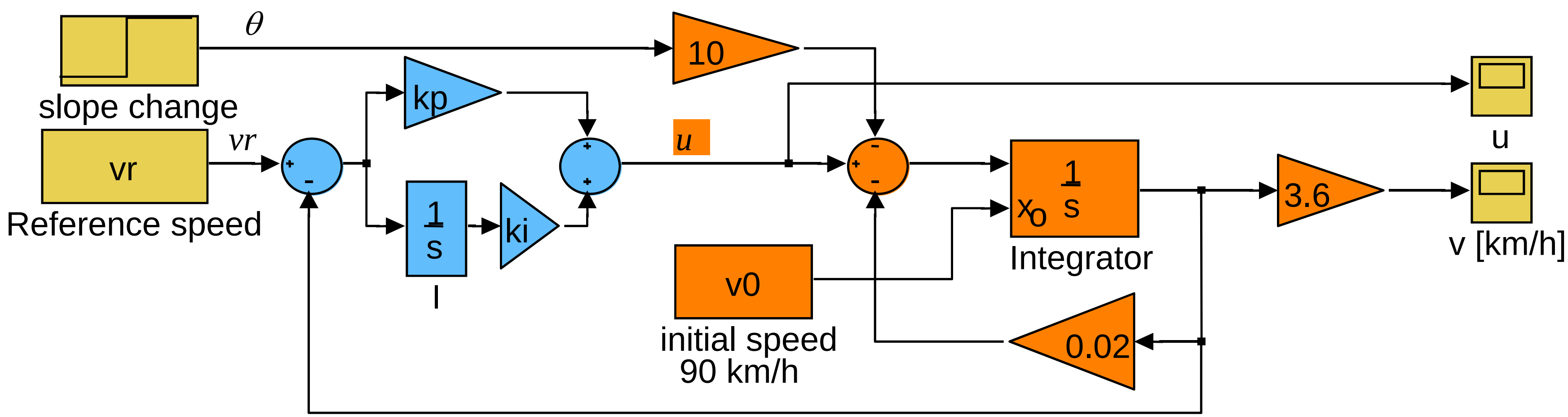
Design problem 1: cruise control (7)

Simulation using MATLAB/SIMULINK:

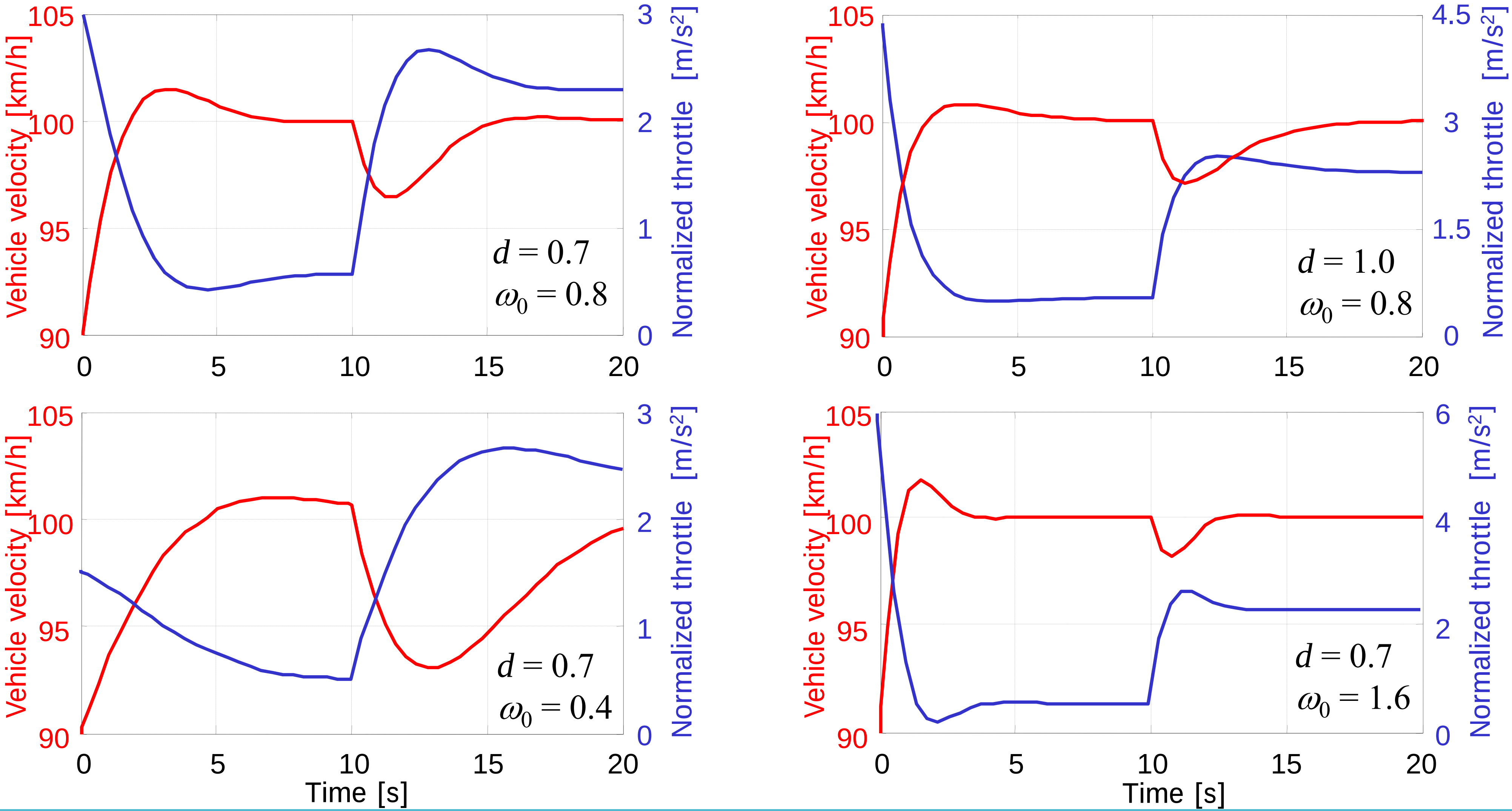
open-loop system



closed-loop with PI controller



Design problem 1: cruise control (8)



Design problem 1: cruise control (9)

What is different between P and PI controlled system (in this special case)?

P control:

- Remaining steady-state control error
- System response is real exponential function, no overshoot (1st order dynamics)
- One common parameter for both the transient and stationary behavior of the closed-loop

PI control:

- No steady-state control error (due to integral action)
- System response is complex exponential function, overshoot possible (2nd order dynamics)
- Stationary behavior independent of controller parameters, closed-loop transient behavior can be separately (oscillation tendency and response dynamic) determined by proper choice of controller parameters

The dynamic order of the closed-loop system is essential for the transient behavior of the controlled system. It is the sum of the dynamic orders of the plant (process) and the controller.

Active excise

Stop and think!

Is the control behavior insensitive (compared to the open loop dynamics) to process parameter changes?

Use the open-loop and closed-loop equations

$$\dot{v} + 0.02v = u - 10\theta \qquad \text{(open-loop)}$$

$$\dot{v} + (0.02 + k_P)v = k_P v_r - 10\theta \qquad \text{(P control)}$$

$$\ddot{e} + (0.02 + k_P)\dot{e} + k_I e = 10\dot{\theta} \qquad \text{(PI control)}$$

for your analysis!

Practical implementation in a vehicle control unit

How will the controller be realized in practice?

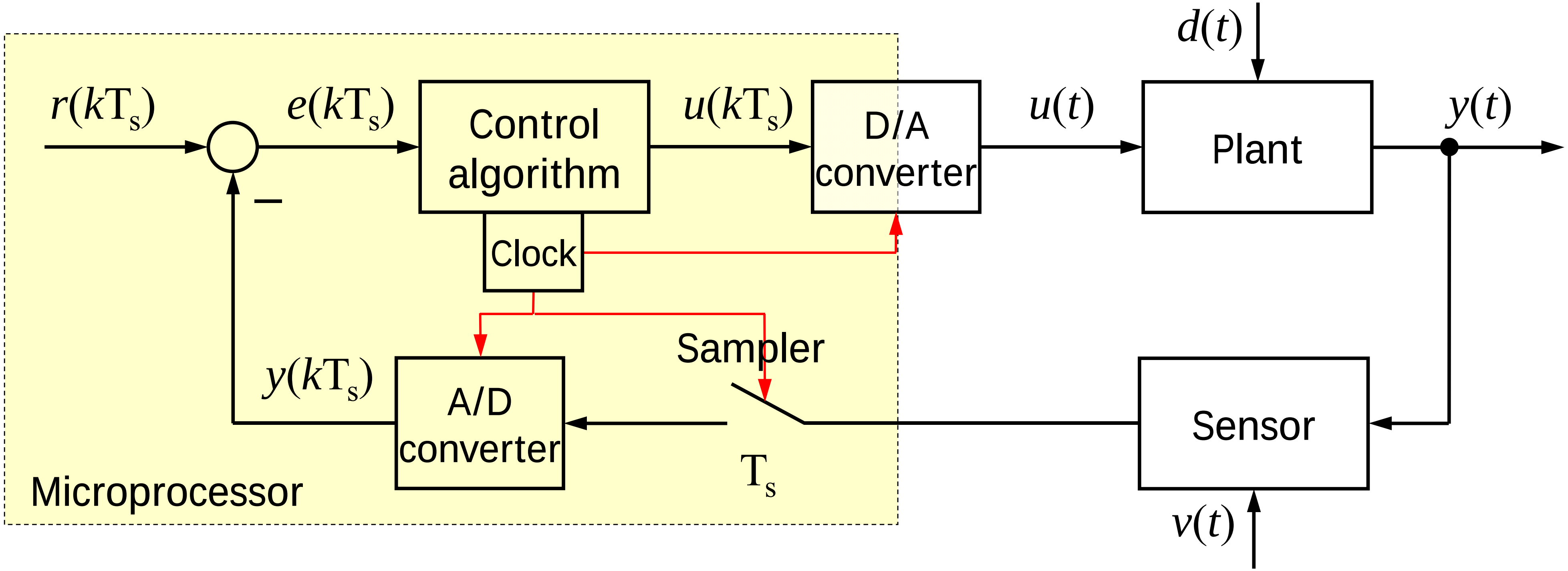
- The controller is a piece of software program (e. g. in the programming language C)
- This program will be compiled in executable codes, loaded into working memory of a processor, and executed clock-controlled in real-time
- The microprocessor is embedded in a vehicle control unit and connected with other components via a field bus system
- The control unit also takes on the control of measurement data acquisition via A/D converter and of the actuator, as well as other computing and communication jobs

What will change in the closed-loop due to the practical implementation?

- Measurement data are available at certain (equidistant) time instants only (sampling)
- The continuous-time differentiation and integration contained in the PID algorithm must be replaced by discrete-time computations (Discretization)
- Die computed discrete-time actuation signal can not be directly, but only via a hold device output to the physical actuation device (D/A converter)

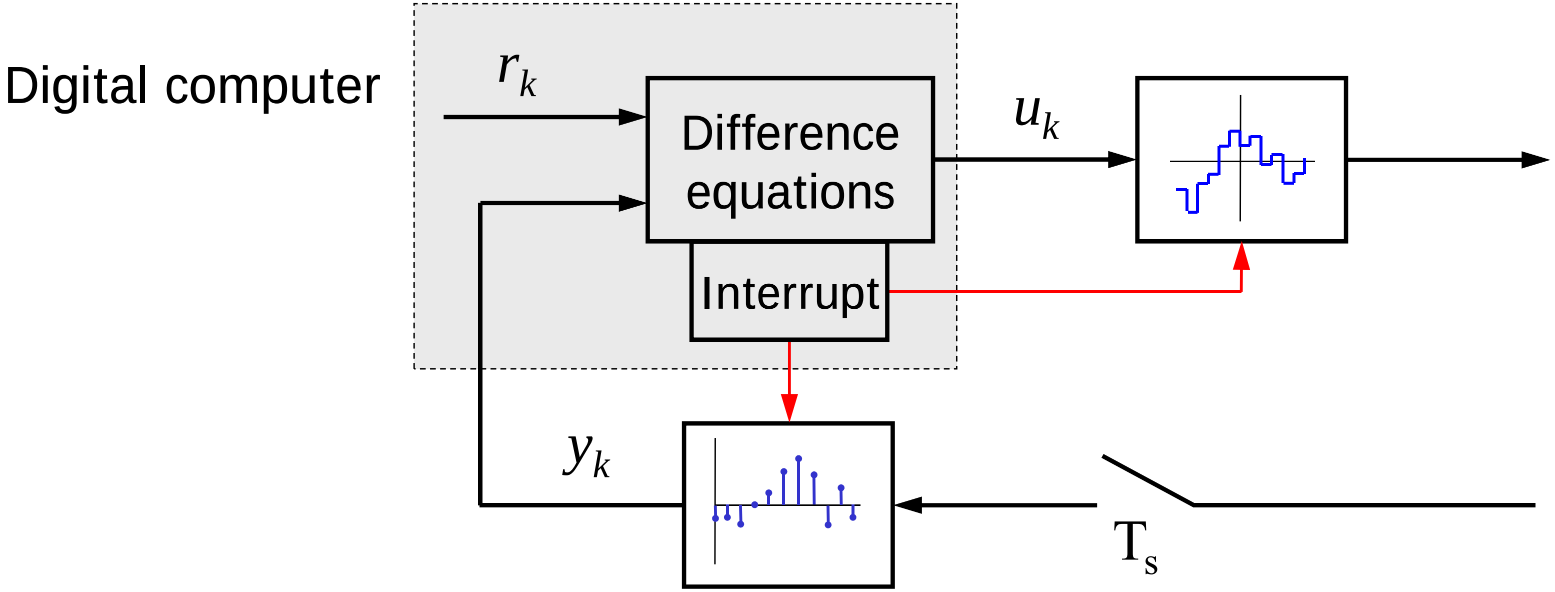
Will the control behavior change then? Think a minute ...

Basic structure of a sampled-data control system



$r(kT_s)$	=	discrete-time reference	$u(t)$	=	real actuation signal
$u(kT_s)$	=	discrete-time actuation signal	$d(t)$	=	contiuous-time disturbance
$y(kT_s)$	=	sensed output	$y(t)$	=	actual output
$e(kT_s)$	=	$r(kT_s) - y(kT_s)$ measured control error	$v(t)$	=	sensor noise

Implementation topology: digitization



- Control algorithm: algebraic recursive equations (difference equations) performing numerical (e.g. Euler's) integration as real-time approximation of differential equations
- A/D converter (timer controlled) repetitively converts a voltage into binary number at sampling instants, *Sampling period*: T_s, *sampling rate*: $1/T_s$
- D/A converter (timer controlled) converts binary numbers to analog voltages followed by usually a *zero-order hold* (ZOH) to generate a continuous signal

A possible „program“ for a PI controller

Time-discretization of a PI controller:

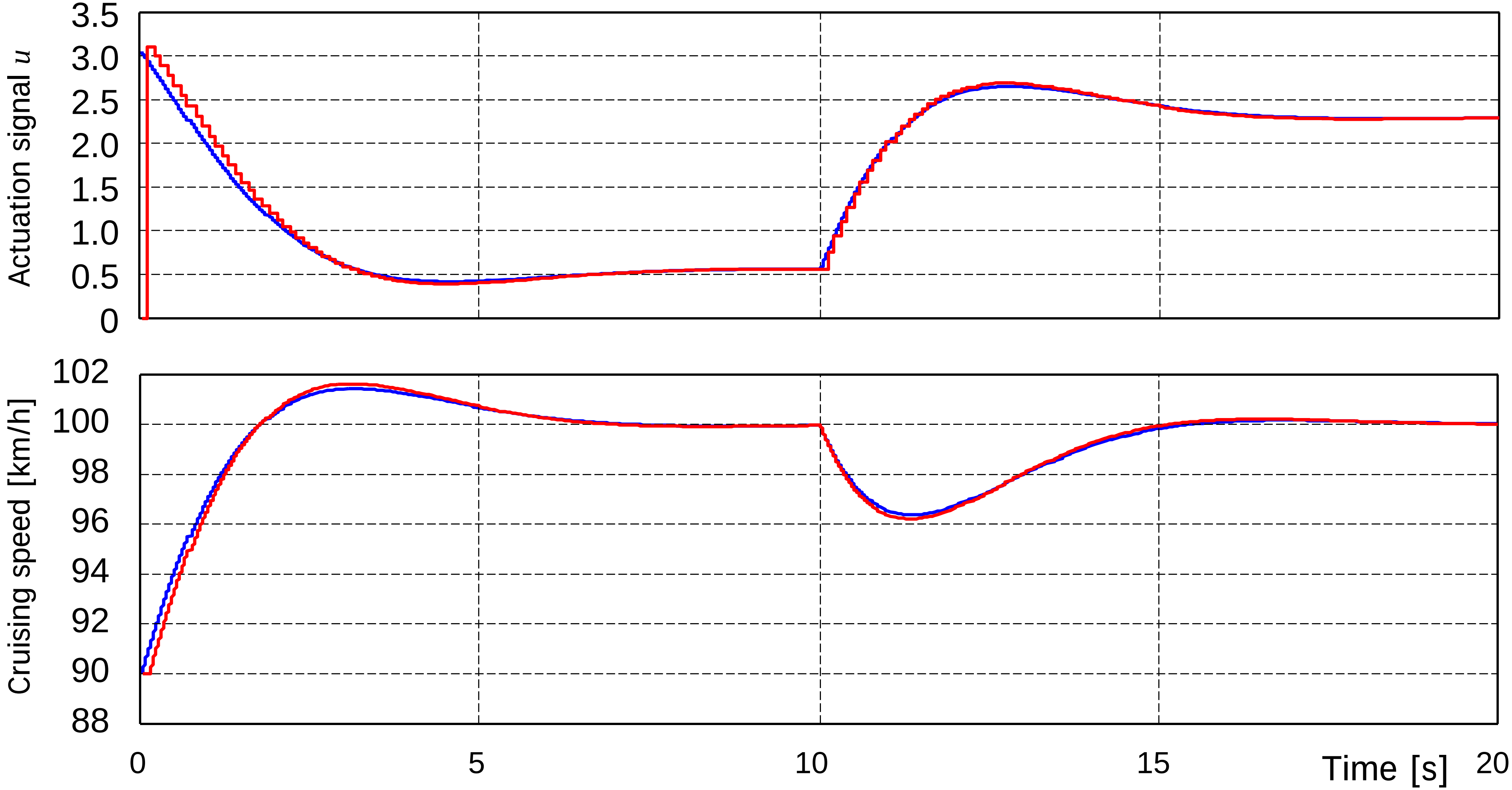
$$u(t) = k_P e(t) + k_I \int_0^t e(\tau) d\tau \xrightarrow{T_S = \text{sampling period}} u_k = u_{k-1} + k_P e_k - k_P e_{k-1} + k_I \, T_s \, e_{k-1}$$

A possible real-time implementation of a discrete-time PI controller:

```
u1 = 0, e1 = 0 (initialization of past values for first loop through)
Define constants:
c1 = Kp, c2 = Ki*ts
READ r and READ A/D to obtain y
e = r – y
u = u1 + C1*e – C1*e1 + C2*e1
OUTPUT u to D/A and ZOH
now compute u1 and e1 for the next loop through
u1 = u, e1 = e
go back to READ when Ts seconds have elapsed since last READ
```

It is ideally assumed that the calculation time between reading of the A/D and writing to the D/A is negligible.

Comparison between analog and digital control



Conclusions

What have we done?

- Construction of the block diagram of the system
- Derivation of process model in the form of ODE (ordinary differential equation)
- Selection of controller ODE
- Understand how closed-loop system behaves
- Selection of controller parameters to achieve desired behavior
- Fine tuning of controller parameters by simulation (or experiment)

How can we generally develop systematic methods to do the control design?

- Derive system differential equations (modeling)
- Eliminate unneeded variables to given relations between input, disturbance and output (modeling)
- Understand the open-loop and closed-loop equations, i.e., analyze the dynamic and steady-state behaviors of the system (system analysis)
- Find proper structure and parameters of the controller using dynamic description (design)
- Validate the results by simulation (validation)