Control Engineering (SC42095)

Lecture 1, 2020

Tamás Keviczky, Azita Dabiri

Delft Center for Systems and Control
Faculty of Mechanical Engineering
Delft University of Technology
The Netherlands

e-mail: t.keviczky@tudelft.nl, a.dabiri@tudelft.nl

Lecture outline

- Information about the course.
- Introduction to computer controlled systems.
- Sampling of continuous-time signals.

Control Engineering (SC42095)

Lecturers: Tamás Keviczky (Mekelweg 2, C-3-310)

Azita Dabiri (Mekelweg 2, C-3-300)

- Lecture recordings from previous years available via Collegerama (search course code/name: SC42095 Digital Control)
- Two online office hours per week:
  - Tuesday 10:45 – 12:30, Zoom (Tamas)
  - Wednesday 8:45 – 10:30, Zoom (Azita)
- Final quiz: January 6, 2020
  - Wednesday 8:45 – 10:30, online via Ans
- Final grade based on project assignment + quiz
  no final exam!

Goals of the course

Main subject:
computer-controlled systems, digital implementation

- Analyze discrete-time systems (analysis).
- Design sampled-data controllers (synthesis).
- Identify implementation issues.
- Apply the design methodology to a simulated process (project assignment with Matlab and Simulink).

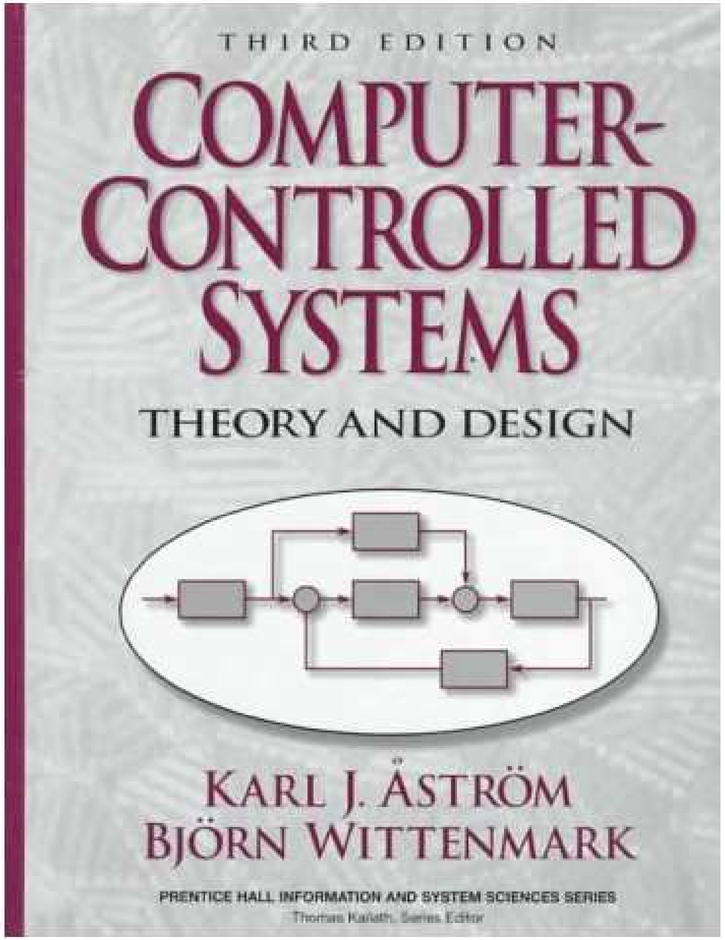
Project assignment

- Purpose: learn to apply the design and analysis techniques and the use of numerical tools
- Procedure:
  - download an assignment from Brightspace,
  - notify the lecturer for confirmation,
  - start with the continuous-time control task,
  - gradually solve the discrete-time problems,
  - write a short report (deadline 12:00, January 8, 2021)
- Your report will be graded and forms the basis of the final mark (80% of the total grade)

Contents of lectures

- Introduction to computer controlled systems
- Sampling of signals and systems (aliasing, discrete-time models, z-transform, zeros and poles)
- Discrete-time state-space and input-output models
- Analysis of discrete-time systems (stability, controllability)
- Design methods
  (state-space, digital PID, linear quadratic, repetitive)
- Observers, stochastic models
  (Luenberger observer, Kalman-filter)

Course material

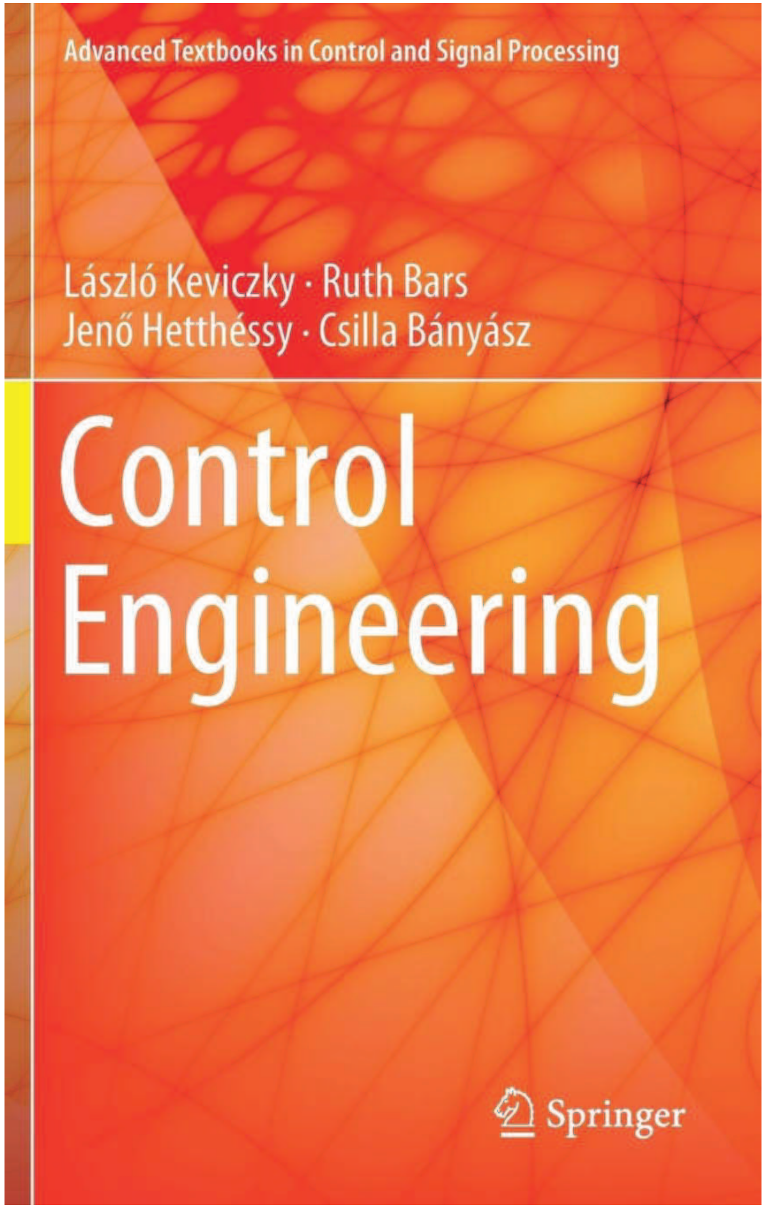


Book:
Aström K.J. and Wittenmark B.:
Computer Controlled Systems
3rd ed., Prentice Hall, 1997.

Slides:
available through Brightspace

MATLAB/Simulink software
available through Brightspace

Supplementary textbook



Book:
L. Keviczky et. al:
Control Engineering, and
Control Engineering: MATLAB Exercises,
Springer, 2019.

Selected chapters on the subject of sampled data control and background material for project assignment (e.g. PID design).

Prerequisites

- Mathematical analysis
- Linear algebra
- Dynamical control systems, state-space theory (SC42015)
- Working knowledge of Matlab and Simulink

Some important concepts to refresh

- Differential equations, Laplace transform.
- Transfer functions, state-space models.
- Poles and zeros, stability, root-locus, . . .
- PID control, system type

Matlab and Simulink

- Elementary Matlab commands (`plot`, `load`, `save`, etc.)
- Control System Toolbox:
  - LTI class (`ss`, `tf`, `zpk`)
  - time-domain and frequency analysis (`step`, `bode`)
  - control design tools (`rlocus`, `place`)
- Simulink (implementing models, simulation methods)

Final Grade

- 80% for the project + 20% for the quiz
- Online Quiz (closed book, closed notes, signup via Google sheets, not in OSIRIS!):
  - about 20 multiple choice questions in one hour
  - testing essential concepts and definitions
- Relevant material: contents of the lectures, handouts and selected parts of the Aström book (download quiz demands).

Important Note

- Some parts of the book will not be discussed in the lectures and are left for self-study (we will not cover all chapters).

- Some material will be discussed in more detail than in the slides using the blackboard, so make sure you watch all the lectures.

Course information on the Web

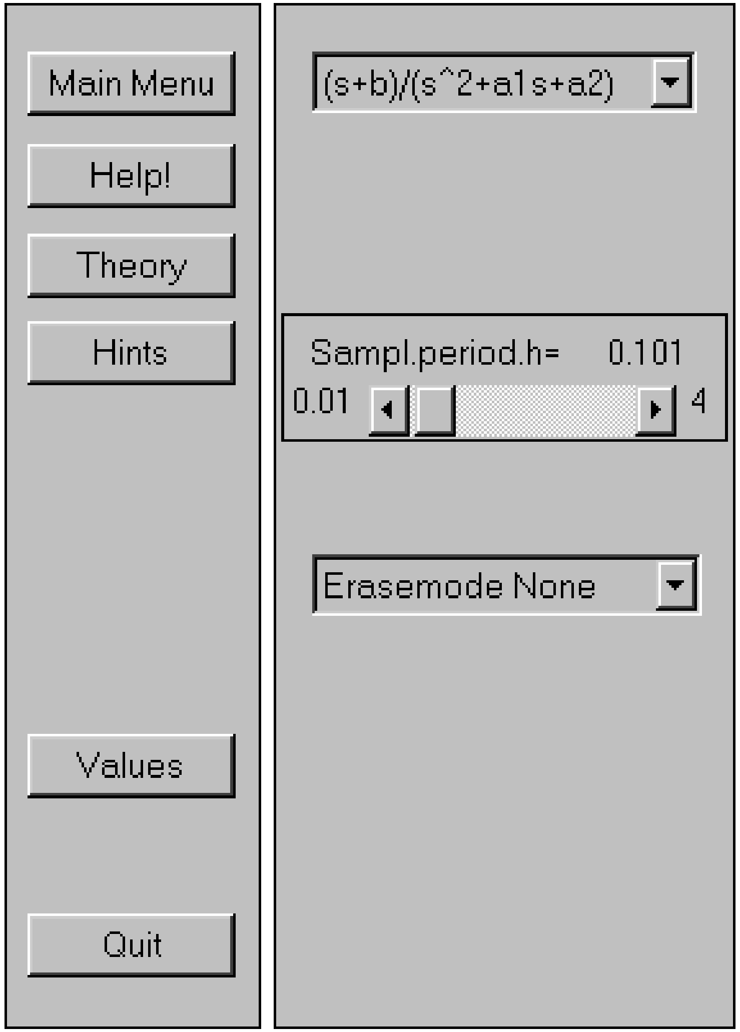
Brightspace

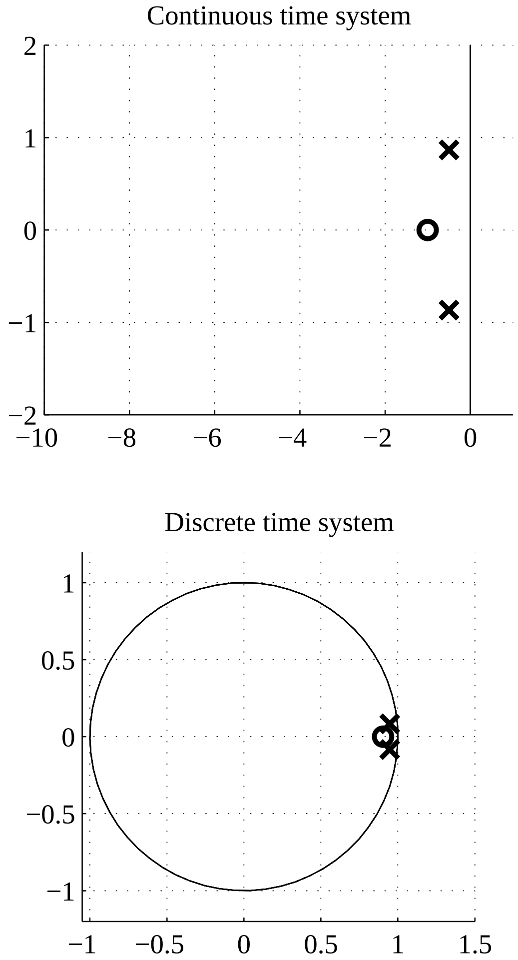
- Basic course information.
- Important dates, messages.
- Solutions manual, errata for the book.
- Exercises, discussion forum, etc.
- Course software (for Matlab and Simulink):
  - Matlab scripts to generate figures in the book: (see how examples are implemented, etc.)
  - CCSDEMO: interactive tool
- Other supplementary references.

Exercises in Brightspace

- Multiple-choice questions.
- Several tests will be available.
- Graded, but no influence on the final mark.
- Should be useful as preparation for the quiz, but it is <u>not</u> a complete set of possible exam questions.

CCS – example



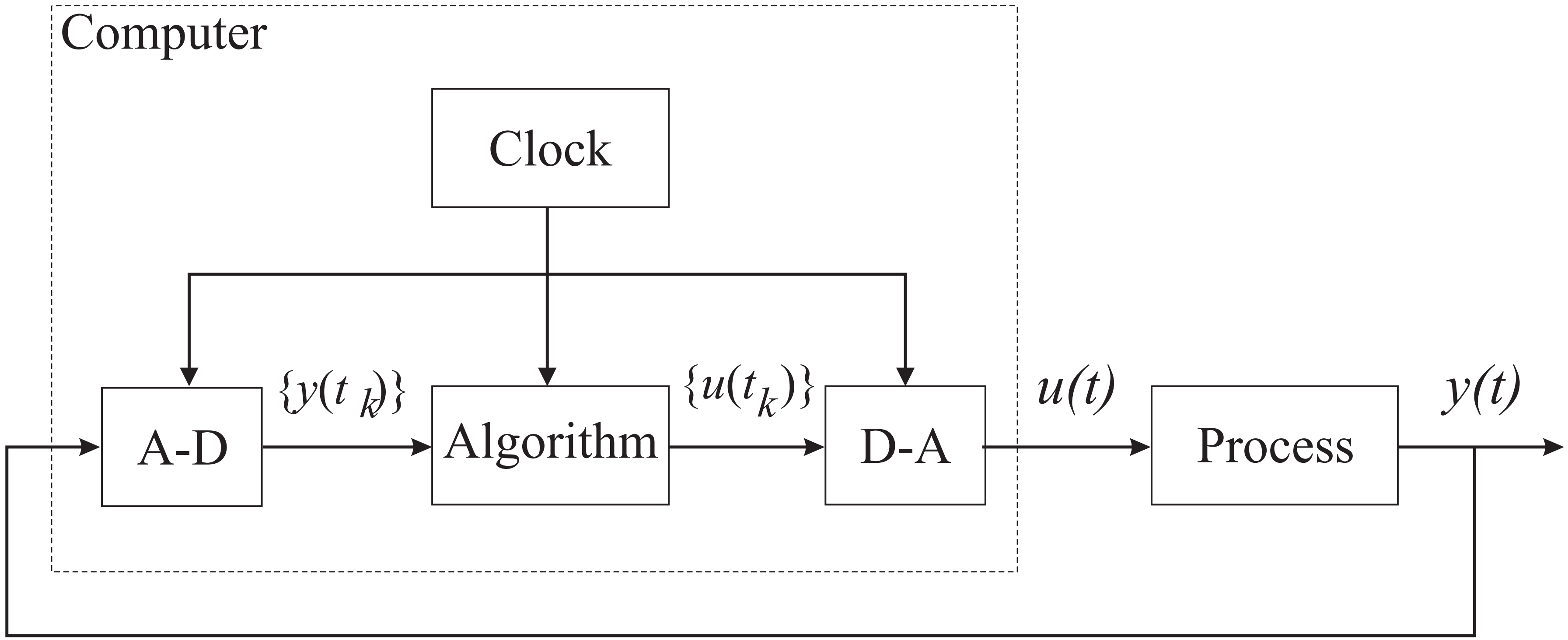


Sampling of

$$\frac{s+b}{s^2+a_1s+a_2}$$

for different sampling periods

Computer-controlled systems



both continuous-time and sampled (discrete-time) signals:
(sampled-data systems) $\rightarrow$ may give some difficulties

Computer-controlled systems

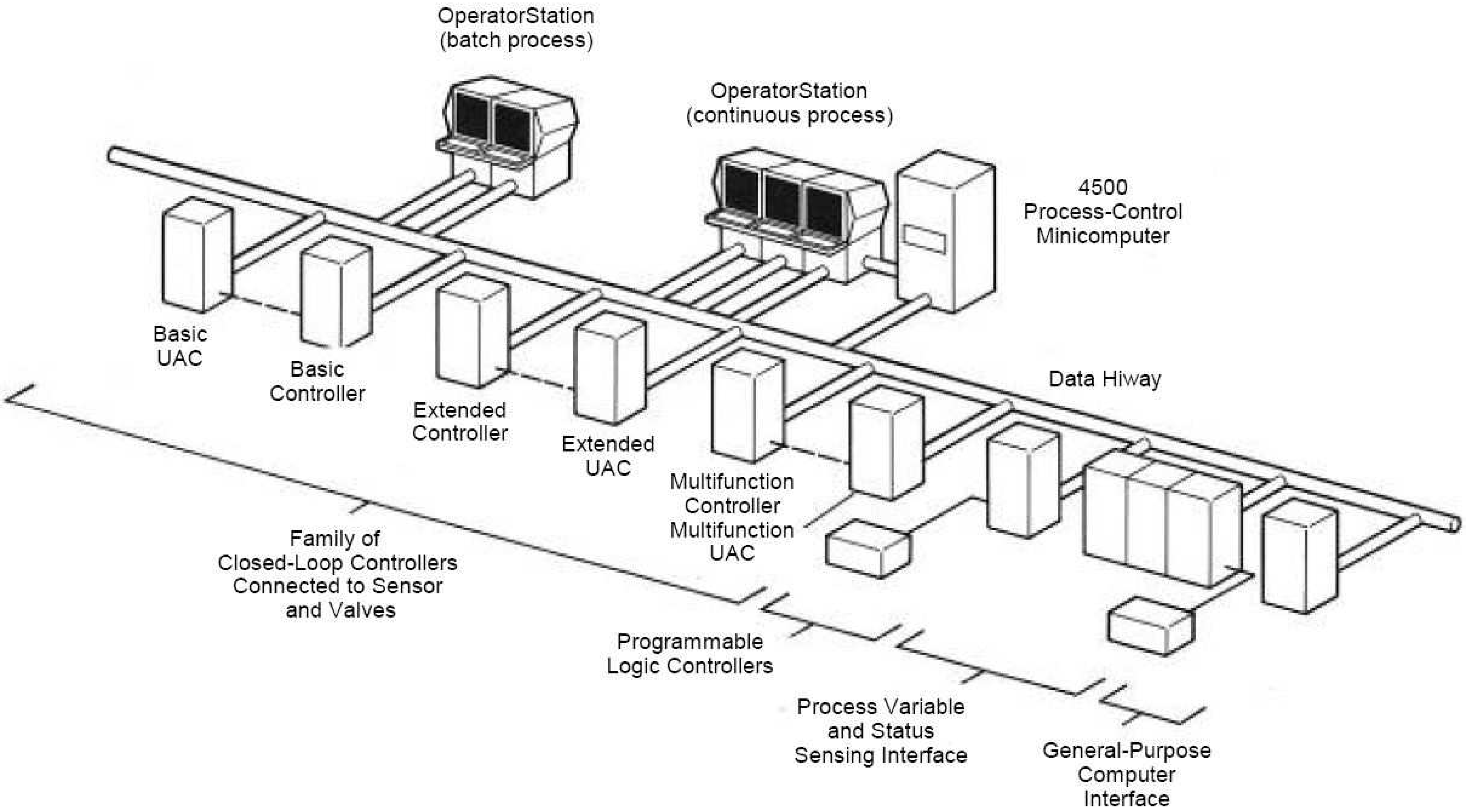
- Common approach: describe process at the sampling instants only, i.e., disregard intersample behavior (discrete-time systems).

- Some systems are inherently sampled, e.g.
  - due to measurement procedure, such as radars, economic systems
  - due to pulsed operation, such as thyristors, combustion engines, biological systems

Development of computer controlled systems

- Pioneering period $\approx$ 1955 (process industry: Texaco, very unreliable with CPU MTBF 50-100h, operator guide, set-point control).
- Direct digital control $\approx$ 1962 (able to control more than 100 variables, still often more expensive than analog control, MTBF 1000h).
- Minicomputers $\approx$ 1967 (16 bits, 124k memory, price still \$10.000 – \$100.000, MTBF 20000h).
- Microcomputers, general use of digital control $\approx$ 1972-80 (\$500, VLSI, standard single-loop controllers, PLCs, microcontrollers).
- Distributed control $\approx$ 1990 (plant-wide control, integration of control, optimization and planning).

Distributed control system (DCS) origins



Honeywell TDC 2000

Distributed control system (DCS) origins

- Honeywell TDC 2000: Development started in 1969, system announced in 1975.
- Several "firsts", and not just for process automation:
  - the first 16-bit microprocessor in a commercial product
  - a "local area network" before the term was known
  - CRT-based operator consoles, a precursor to the GUI
- Basic TDC 2000 controller is still running many refineries world-wide (e.g., Shell Pernis).

Development of theory

- Sampling Theorem (1949)
- Difference Equations (1948)
- Numerical Analysis
- Transform Methods (1947)
- State-space Theory (1958)
- Optimal Control (1957)
- Systems Theory (1969)
- System Identification (1971)
- Adaptive Control (1973)
- Automatic Tuning (1995)

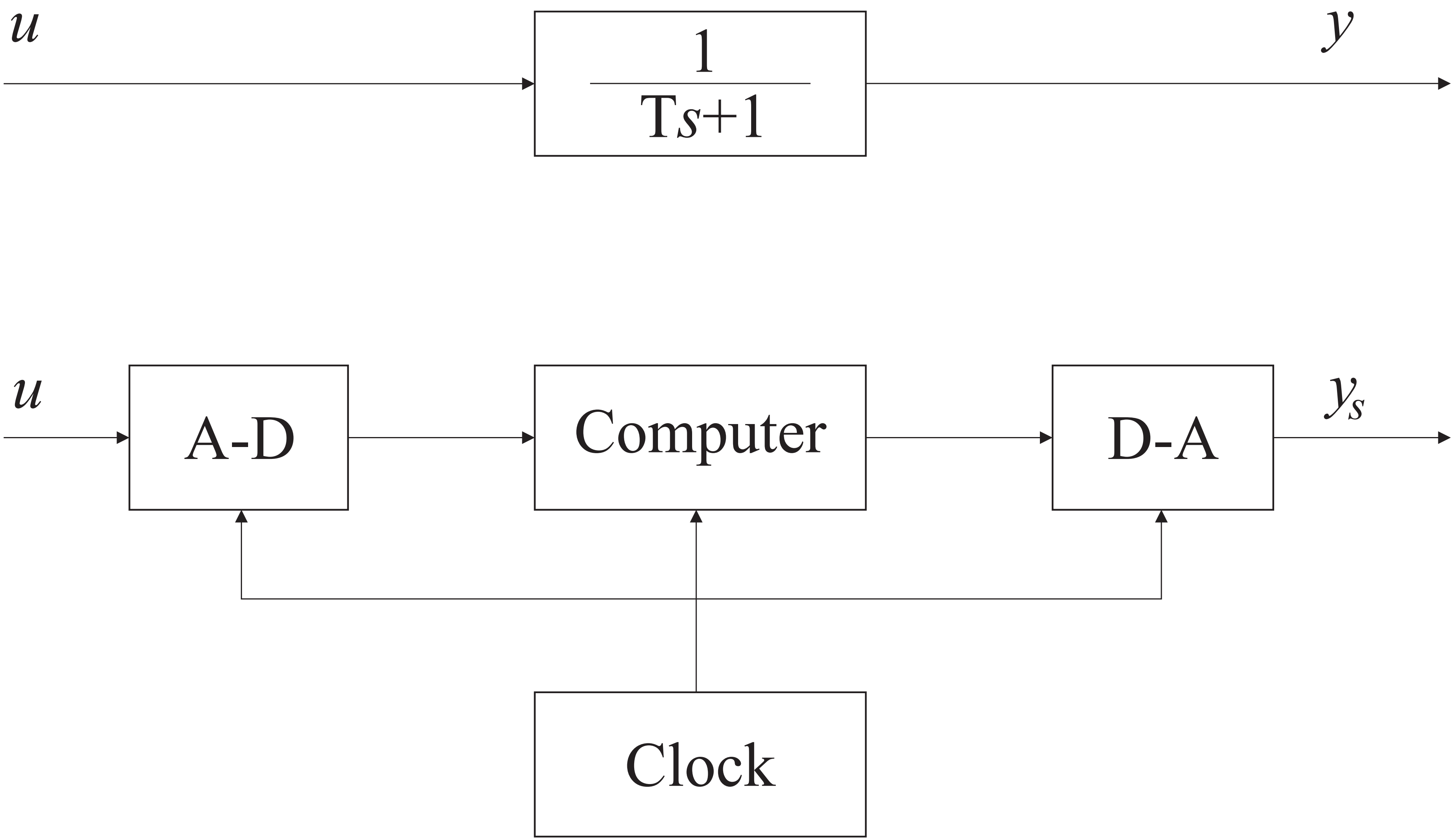
Advantages of computer control

- cheap hardware, no drifting, no ageing
- extra control performance (example later on)
- include models, nonlinearities
- signal processing, adaptation ('learning')
- complex calculations, logical functions, sequencing, flexibility
- integrated (plant-wide) control, user interface, data storage

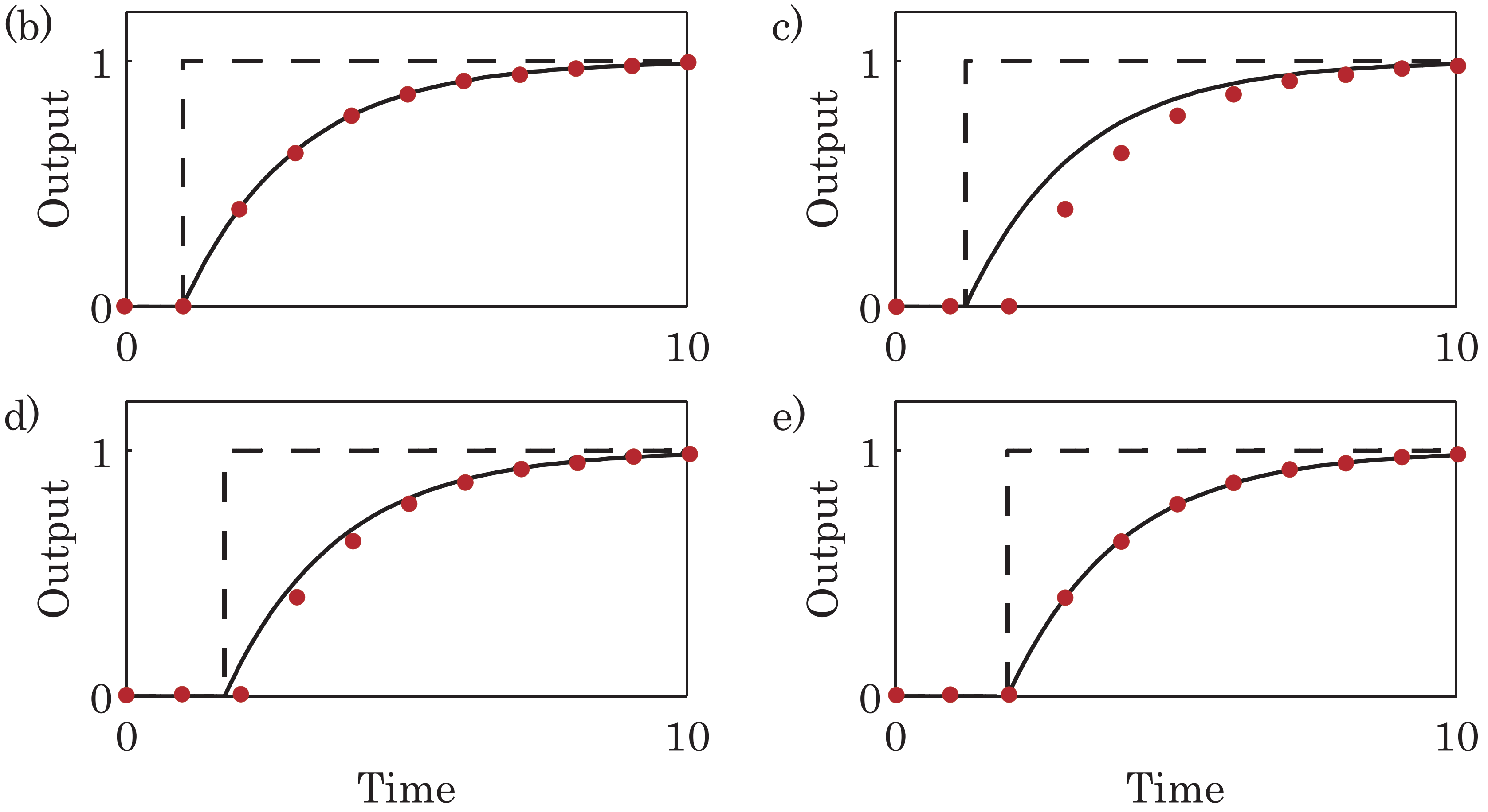
Limitations of computer control

- limited performance of computers (speed)
  - real time aspects
  - large-scale systems, networks
- robustness, safety (faults)
  - maintenance (e.g., cars)
  - safety-critical applications (e.g., aircraft, reactors)
- sampling $\rightarrow$ new phenomena (extra caution)
  - time-dependence
  - influence of sampling period
  - sampling of high-frequency signals (noise)

Time-varying behavior



Time-varying behavior – cont'd



dashed line: input step (u), solid line: response of continuous filter (y), dotted line: response of digital filter (y_s)

Influence of sampling period

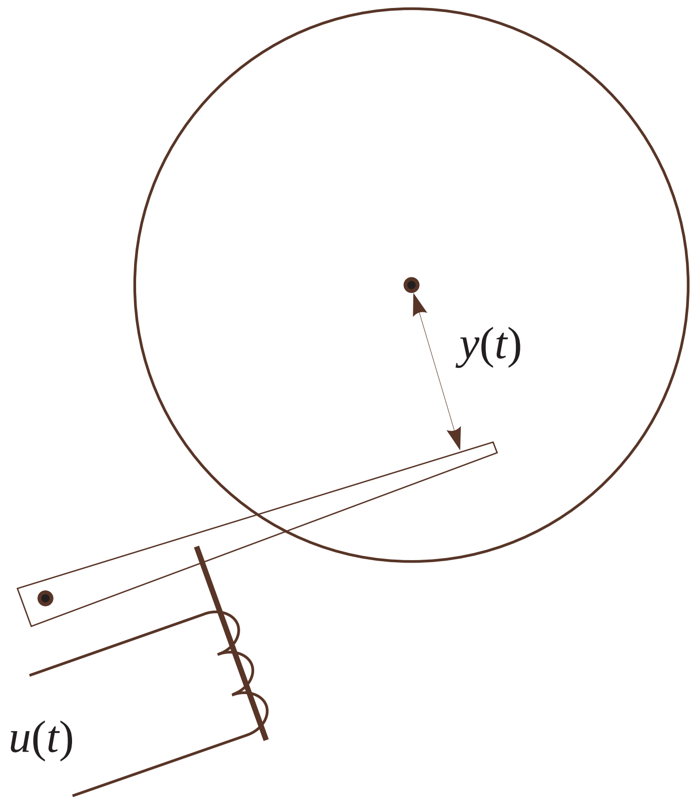
A naive approach:

1. Design continuous-time controller.
2. Discretize this controller.
3. Implement on a computer.

Influence of sampling period – cont'd

Example: control of a disk drive arm (double integrator):

$$G(s) = \frac{Y(s)}{U(s)} = \frac{k}{Js^2}$$



1. Design continuous-time controller

$$U(s) = \frac{bK}{a}U_c(s) - K\frac{s+b}{s+a}Y(s) \tag{1}$$

with

$$\begin{aligned} a &= 2\omega_0 \\ b &= \omega_0/2 \\ K &= 2J\omega_0^2/k \end{aligned}$$

a closed loop characteristic polynomial is obtained

$$P(s) = s^3 + 2\omega_0 s^2 + 2\omega_0^2 s + \omega_0^3$$

1. Design continuous-time controller – cont'd

From (1) after elementary manipulations:

$$U(s) = K\left(\frac{b}{a}U_c(s) - Y(s) + \frac{a-b}{s+a}Y(s)\right)$$

$$\boxed{\begin{aligned} u(t) &= K\left(\frac{b}{a}u_c(t) - y(t) + x(t)\right) \\ \frac{dx(t)}{dt} &= -ax(t) + (a-b)y(t) \end{aligned}}$$

2. Discretize continuous-time controller

$$u(t) = K\left(\frac{b}{a}u_c(t) - y(t) + x(t)\right)$$

$$\frac{dx(t)}{dt} = -ax(t) + (a-b)y(t)$$

Approximate the derivative:

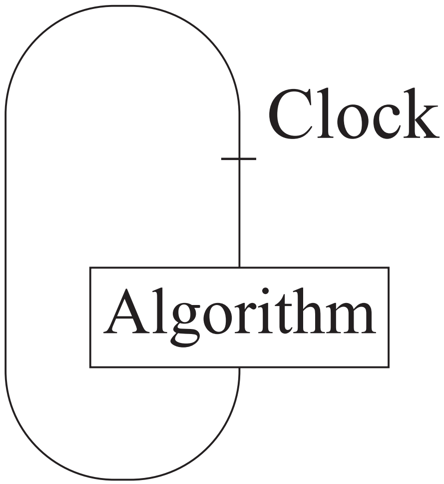
$$\frac{x(t+h) - x(t)}{h} = -ax(t) + (a-b)y(t)$$

Discrete-time controller:

$$\boxed{\begin{aligned} u(t_k) &= K\left(\frac{b}{a}u_c(t_k) - y(t_k) + x(t_k)\right) \\ x(t_k+h) &= x(t_k) + h\Big[(a-b)y(t_k) - ax(t_k)\Big] \end{aligned}}$$

3. Implement discrete-time controller

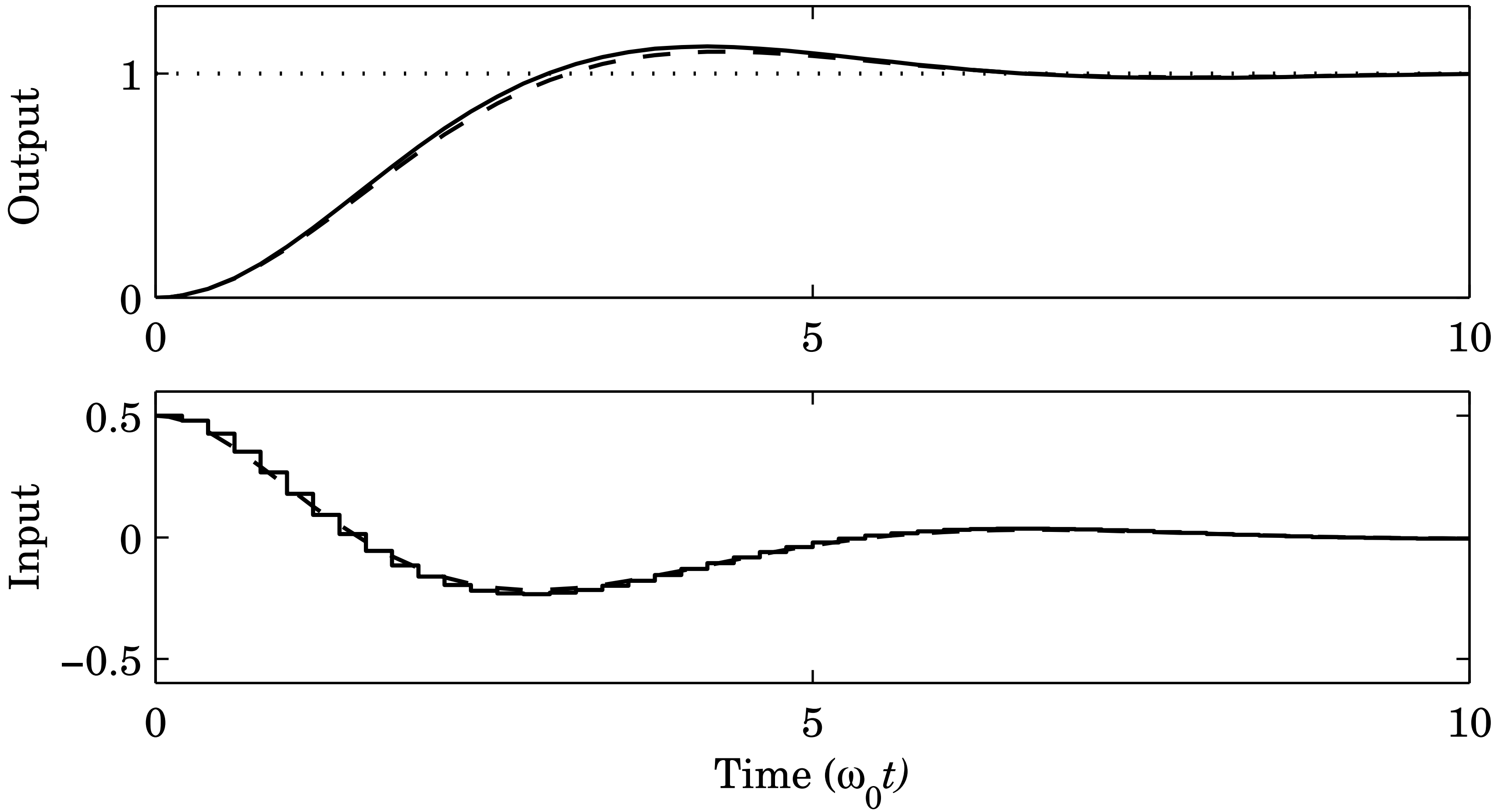
$$
\begin{aligned}
u(t_k) &= K\left(\frac{b}{a}u_c(t_k) - y(t_k) + x(t_k)\right) \\
x(t_k + h) &= x(t_k) + h\Big[(a-b)y(t_k) - ax(t_k)\Big]
\end{aligned}
$$




```
y:= adin                    {read process value}
u:= K*(b/a*uc-y+x)
dout(u)                     {output control signal}
x:= x+h*((a-b)*y - a*x)
```

Sampling period $h = 0.2/\omega_0$

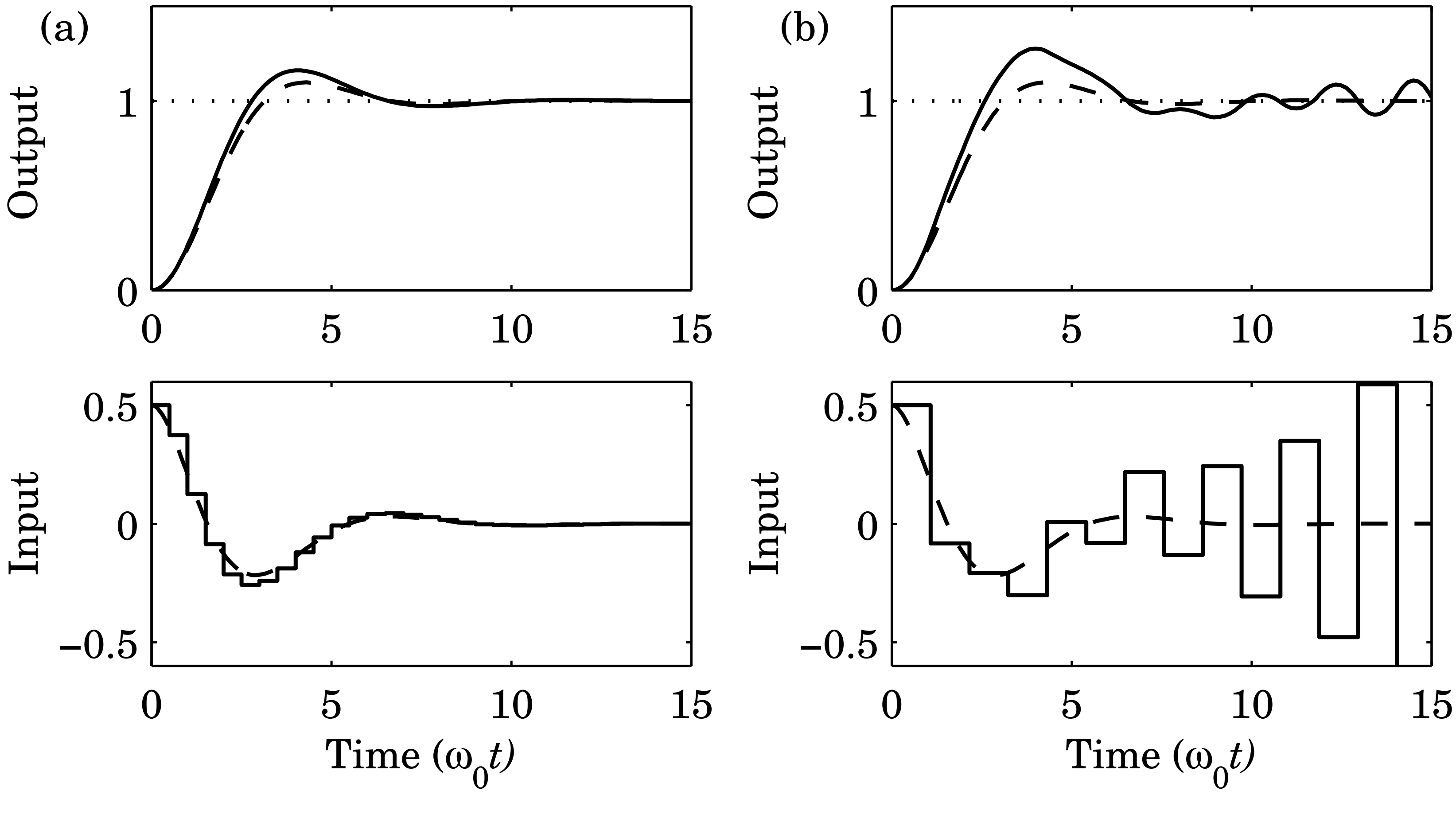
Performance Comparison



solid line: digital controller, dashed line: analog controller

Longer sampling period

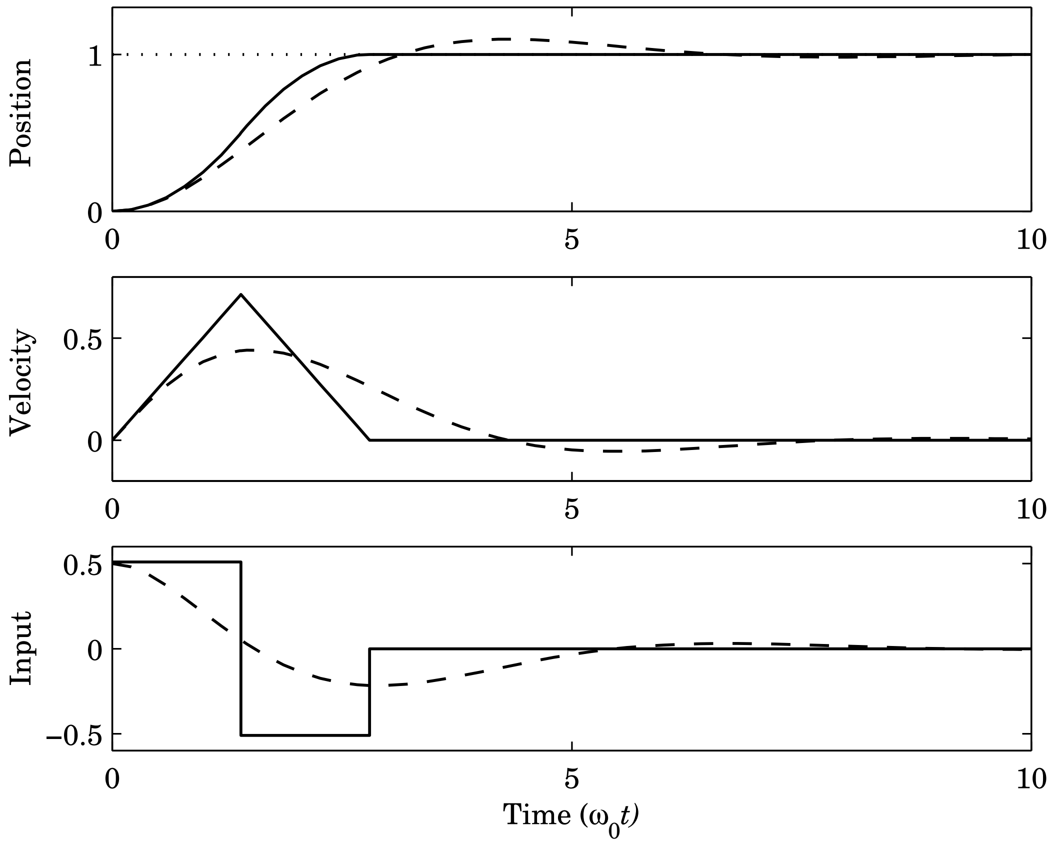
a) $h = 0.5/\omega_0$, b) $h = 1.08/\omega_0$



solid line: digital, dashed line: analog

Digital controllers can do better

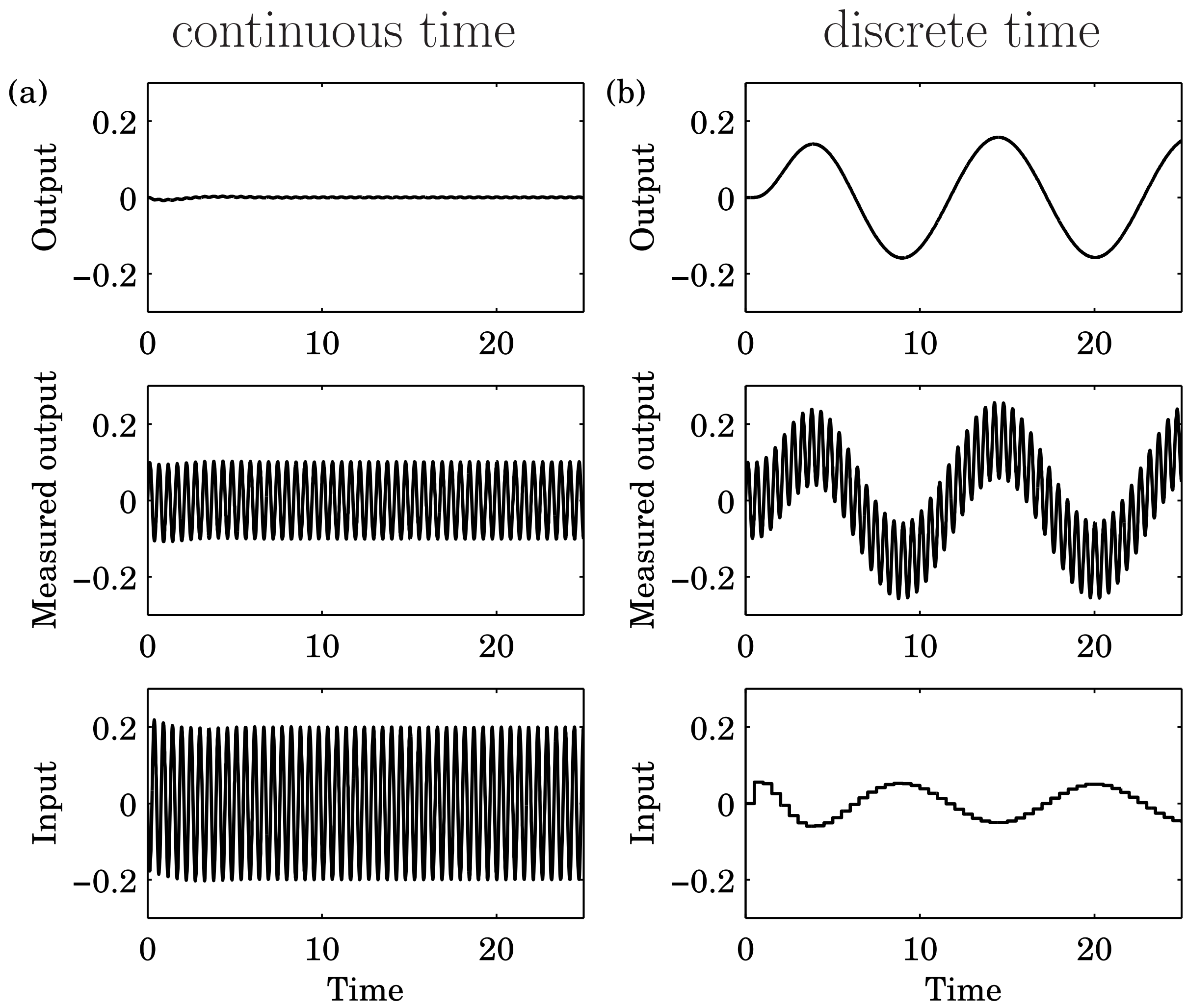
$u(t_k) = t_0 u_c(t_k) + t_1 u_c(t_{k-1}) - s_0 y(t_k) - s_1 y(t_{k-1}) - r_1 u(t_{k-1})$
$h = 1.4/\omega_0 \quad (!)$



solid line: digital, dashed line: analog

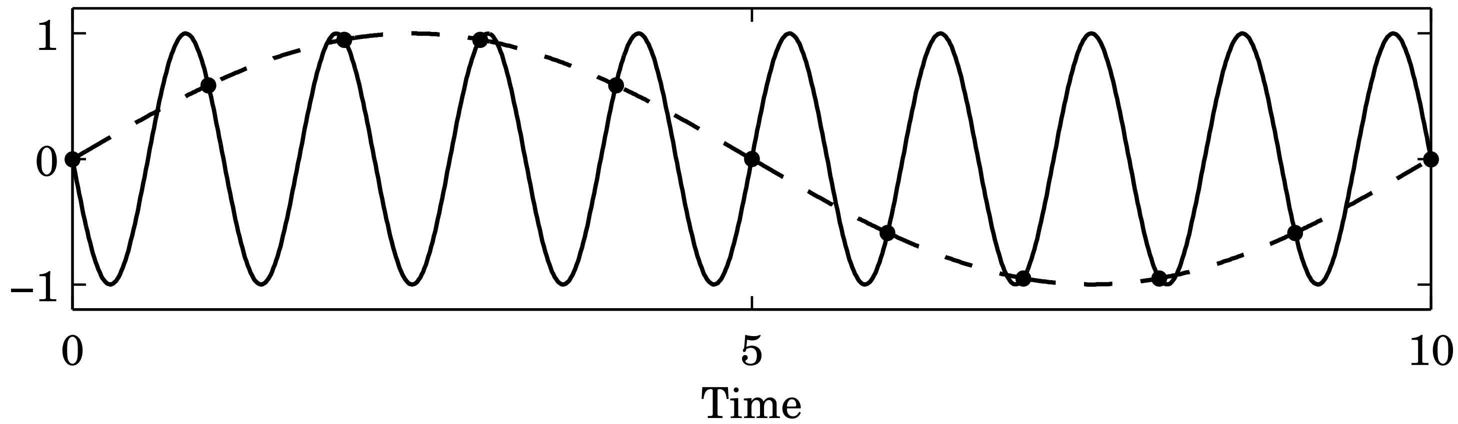
Sampling of measurement noise

Double integrator, $h = 0.5$, measurement noise $0.1\sin(12t)$



Sampling creates signals with new frequencies!

Aliasing



Sampling of two or more harmonic signals with *different* frequencies may result in *the same* discrete time signal. This is called aliasing.

$$\omega_{sampled} = |\omega \pm n\omega_s|, \quad n \text{ integer}$$

with

ω ... frequency of continuous-time signal

$\omega_s = 2\pi/h$ ... sampling frequency

Summary

- Advantages and drawbacks of computer control.
- Short sampling interval may be OK, but we can do better by using discrete-time design.
- Alias frequencies

$$\omega_{sampled} = |\omega \pm n\omega_s|$$