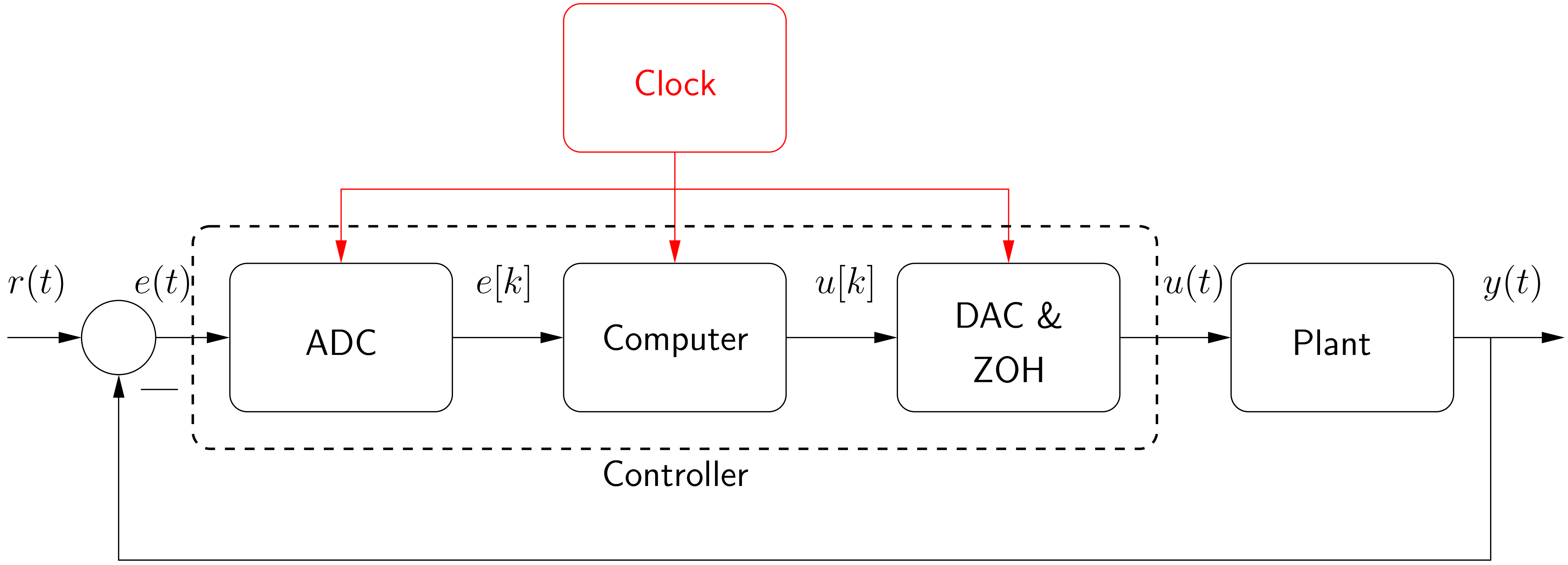
Introduction to Discrete-Time Control Systems

Overview

- Computer-Controlled Systems
- Sampling and Reconstruction
- A Naive Approach to Computer-Controlled Systems
- Deadbeat Control
- Is there a need for a theory for computer-controlled systems?

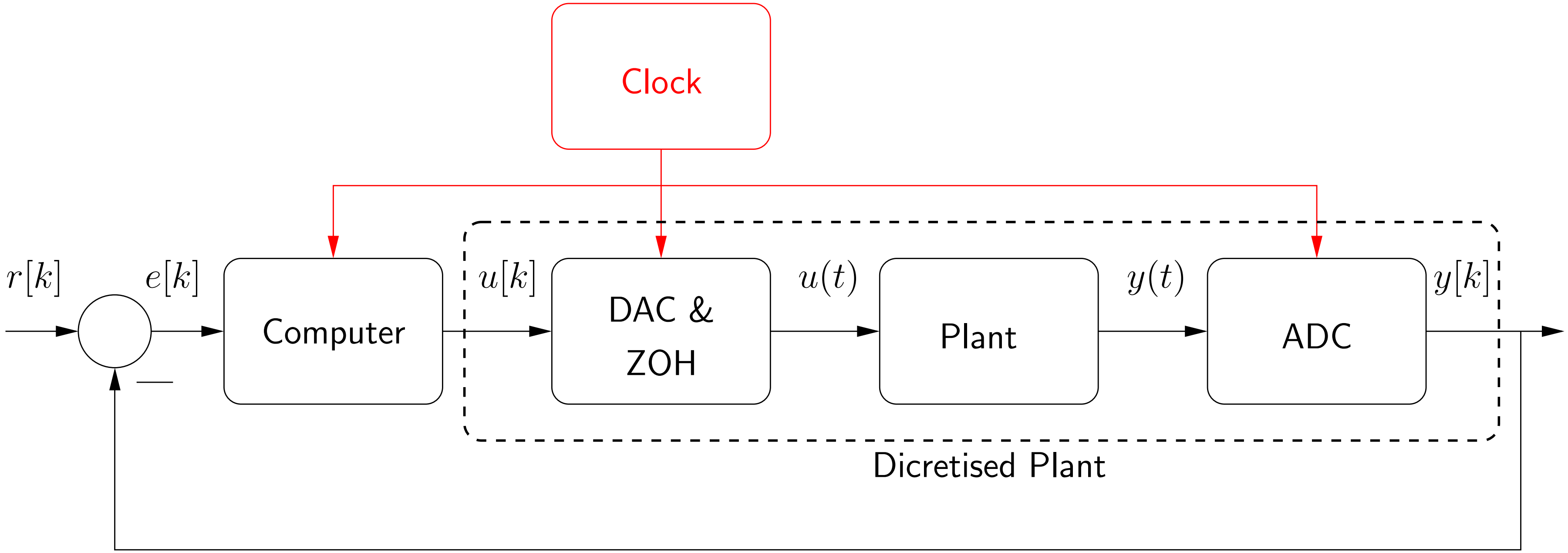
Computer-Controlled Systems

- Implementation of controllers, designed in continuous-time, on a micro-controller or PC (digital realisation of an 'analogue' controller)



ADC - Analog-Digital-Converter (includes sampler), DAC - Digital Analogue Converter,
ZOH - Zero Order Hold

- Direct design of a digital controller for a discretised plant
  - or for identified time-discrete models
  - or for inherently sampled systems (e.g. control of neuro-prosthetic systems)
  - enables larger sampling times compared to the digital realisation of ‘analogue’ controllers
  - enables other features that are not possible in continuous time control (e.g. deadbeat control, repetitive control)



- Components
  - A-D converter (ADC) and D-A converter (DAC)
  - Algorithm
  - Clock
  - Plant
- Contains both continuous and sampled, or discrete-time signals
  $\longrightarrow$ *sampled-data systems* (synonym to computer-controlled system)
- Mixture of signals makes description and analysis sometimes difficult.
- However, in most cases, it is sufficient to describe the behaviour at sampling instants.
  $\longrightarrow$ *discrete-time systems*

- ADC samples a continuous function $f(t)$ at a fixed *sampling period* $\varDelta$
  $\rightsquigarrow$ sequence $\{f[k]\}$ of numbers
- $\{f[k]\}$ denotes a sequence $f[0], f[1], f[2], \ldots$

$$f[k] = f(k\varDelta), \quad k = 0, 1, 2, \ldots$$

- Sampling times / sampling instants $k\varDelta$ or short only k if sampling period is constant.
- Quantisation effects by the ADC (due to limited resolution) are not taken into account at the moment.
- DAC and Zero-Order-Hold approximately reconstructs a continuous from a sequence of numbers.

Sampling

- Sampling frequency needs to be large enough in comparison with the maximum rate of change of $f(t)$.
- Otherwise, high frequency components will be mistakenly interpreted as low frequencies in the sampled sequence.

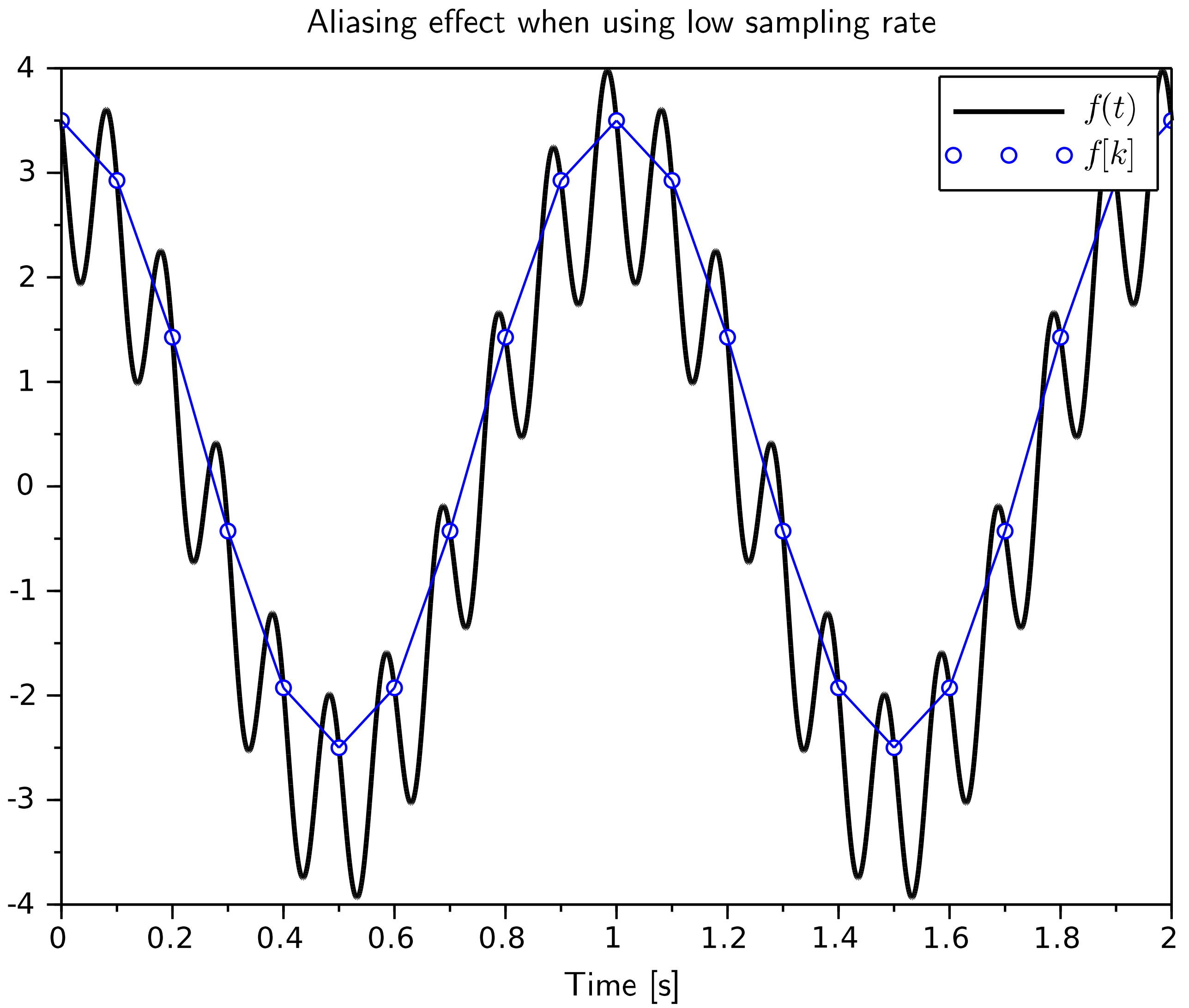
Example:

$$f(t) = 3\cos 2\pi t + \cos\left(20\pi t + \frac{\pi}{3}\right)$$

for $\Delta = 0.1\,\mathrm{s}$ we obtain

$$\begin{aligned} f[k] &= 3\cos(0.2\pi k) + \cos\left(2\pi k + \frac{\pi}{3}\right) \\ f[k] &= 3\cos(0.2\pi k) + 0.5 \end{aligned}$$

The high frequency component appears as a signal of low frequency (here zero). This phenomenon is known as *aliasing*.


Aliasing effect when using low sampling rate
4
3
2
1
0
-1
-2
-3
-4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time [s]
$f(t)$
$f[k]$

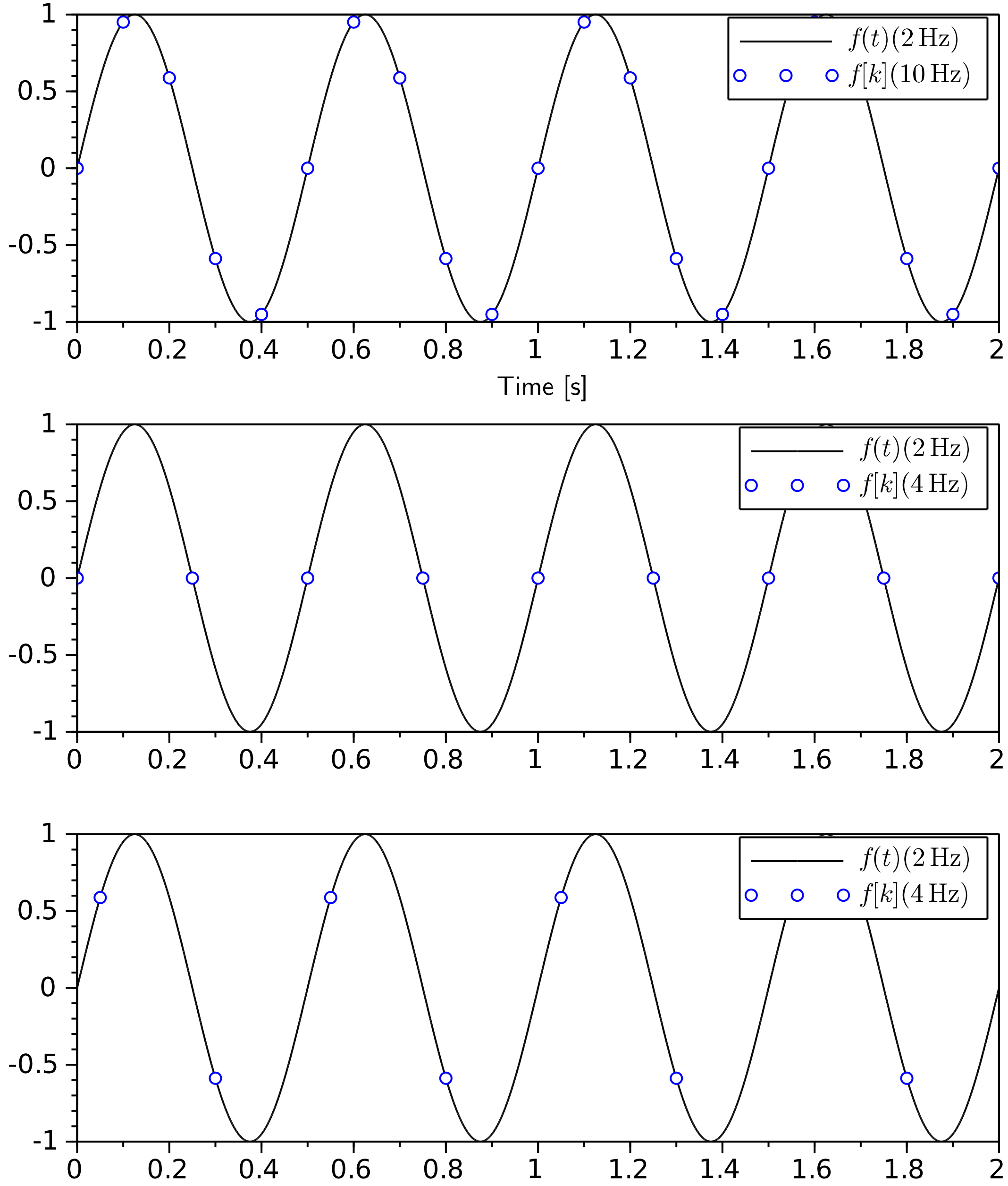
SHANNON'S SAMPLING THEOREM

A continuous-time signal with a spectrum that is zero outside the interval $(-\omega_0, \omega_0)$ is given uniquely by its values in equidistant points if the sampling angular frequency $\omega_s = 2\pi f_s$ in rad/s is higher than $2\omega_0$.

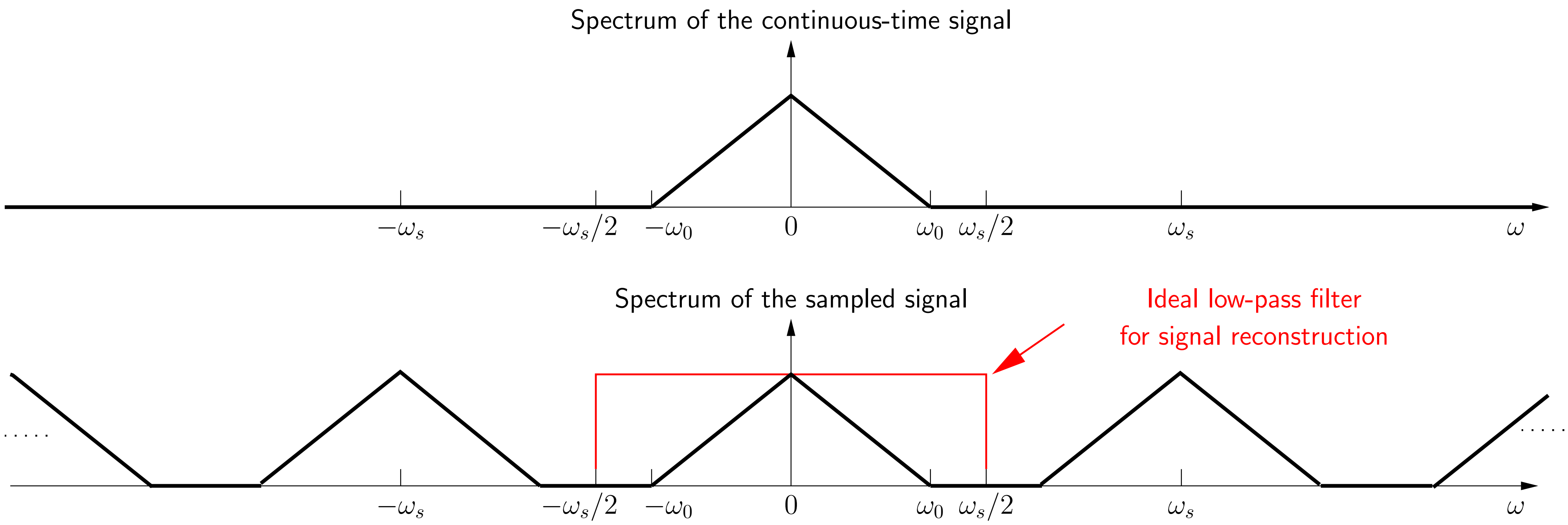
The continuous-time signal can be reconstructed from the sampled signal by the interpolation formula

$$f(t) = \sum_{k=-\infty}^{\infty} f[k] \frac{\sin(\omega_s(t - k\Delta)/2)}{\omega_s(t - k\Delta)/2}$$

- The frequency $\omega_N = \omega_s/2$ plays an important role. This frequency is called the *Nyquist frequency*.
- A typical rule of thumb is to require that the sampling rate is 5 to 10 times the bandwidth of the system.
- The *Shannon reconstruction* given above is not useful in control applications as the operation is non-causal requiring past and future values.

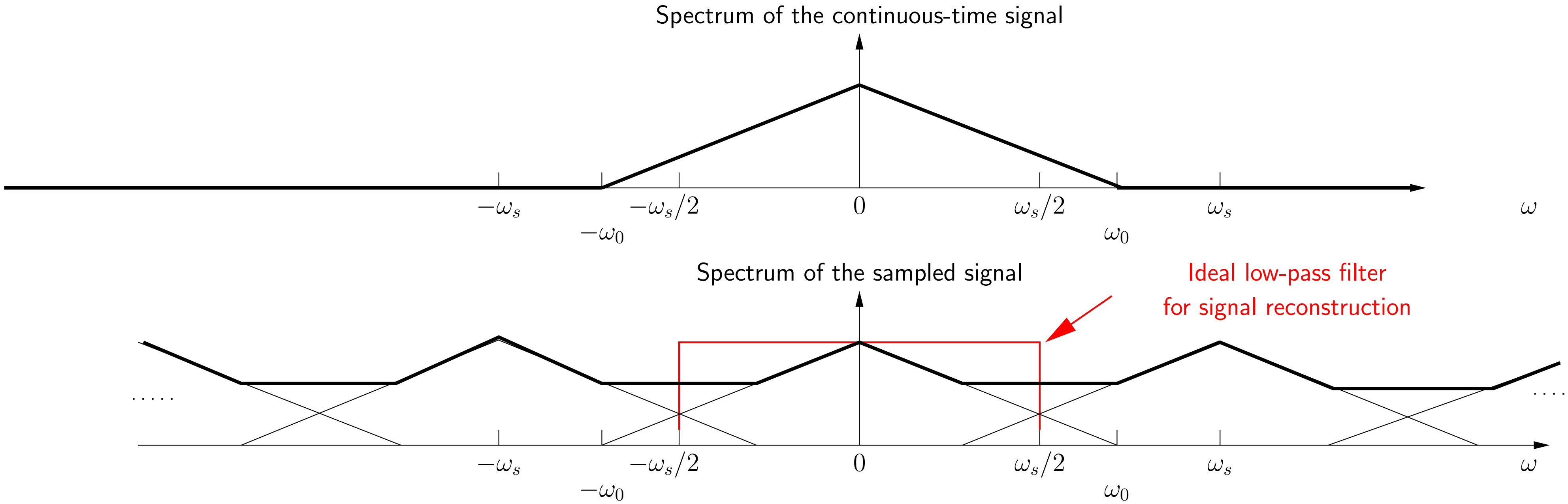

1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time [s]
$f(t)(2\,\mathrm{Hz})$
$f[k](10\,\mathrm{Hz})$
$f(t)(2\,\mathrm{Hz})$
$f[k](4\,\mathrm{Hz})$
$f(t)(2\,\mathrm{Hz})$
$f[k](4\,\mathrm{Hz})$

Sprectra of continuous-time band-limited signal and sampled signal for $\omega_s > 2\omega_0$ $(\omega_N > \omega_0)$.



- Original signal could be reconstructed by ideal low-pass filter.
- Zero order hold is a not so good approximation of an ideal low-pass filter, but simple to implement and therefore often used (risk that higher frequencies created by sampling remain in the control system).

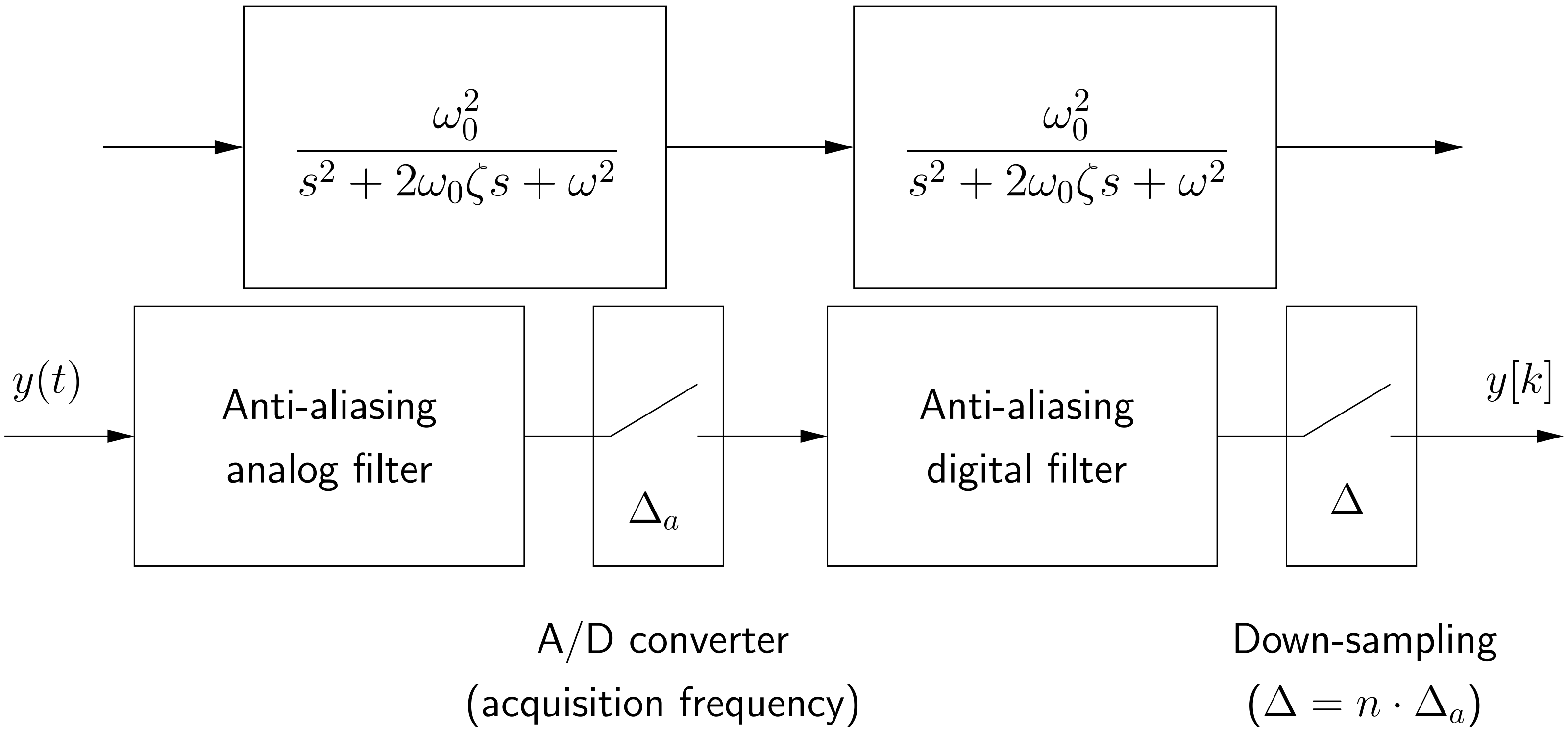
Sprectra of continuous-time band-limited signal and sampled signal for $\omega_s < 2\omega_0$ $(\omega_N < \omega_0)$.



- Original signal cannot be reconstructed filter due to aliasing.
- A signal with frequency $\omega_d > \omega_N$ appears as signal with the lower frequency $(\omega_N - \omega_d)$ in the sampled signal.

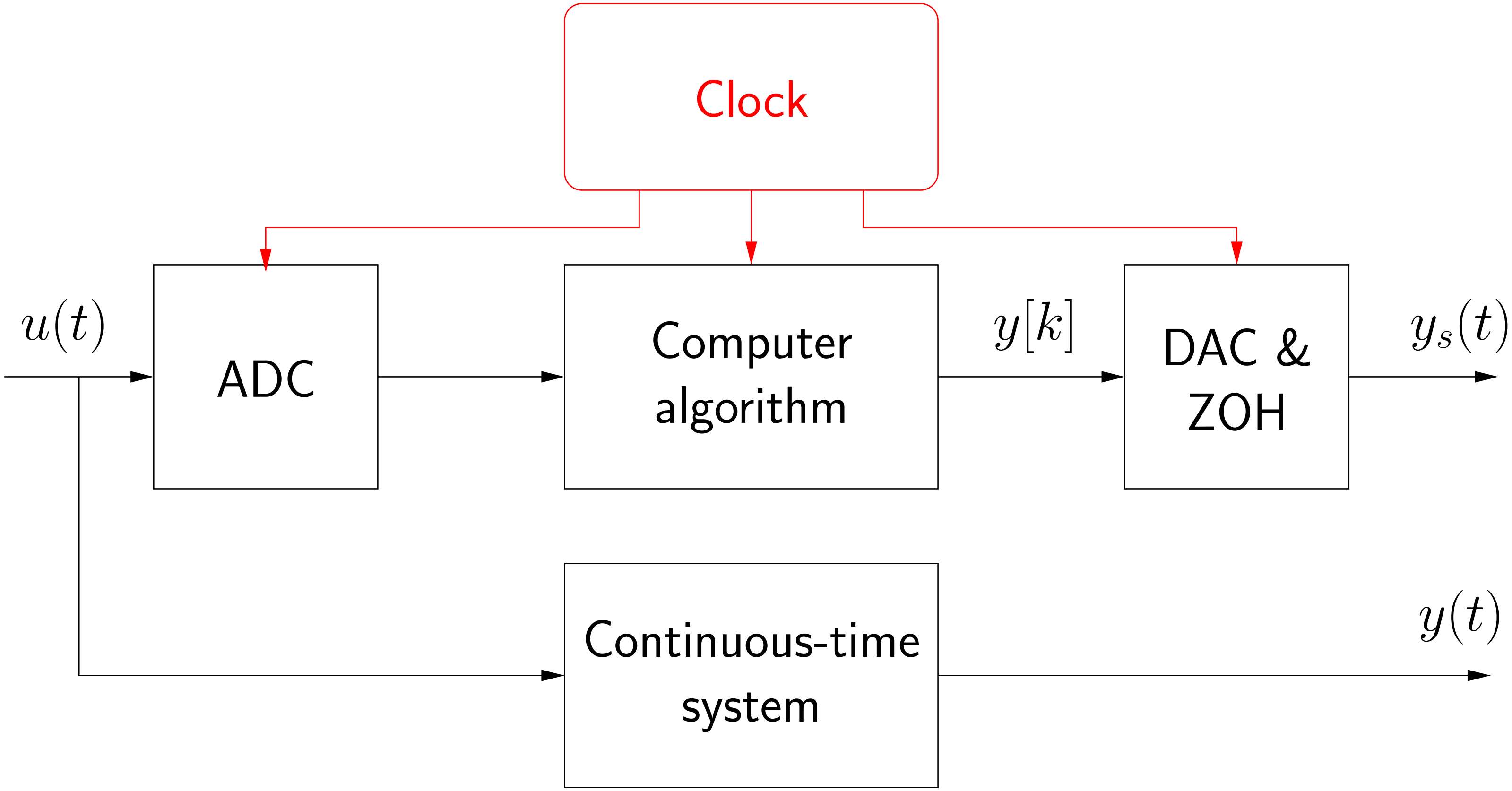
Preventing Aliasing

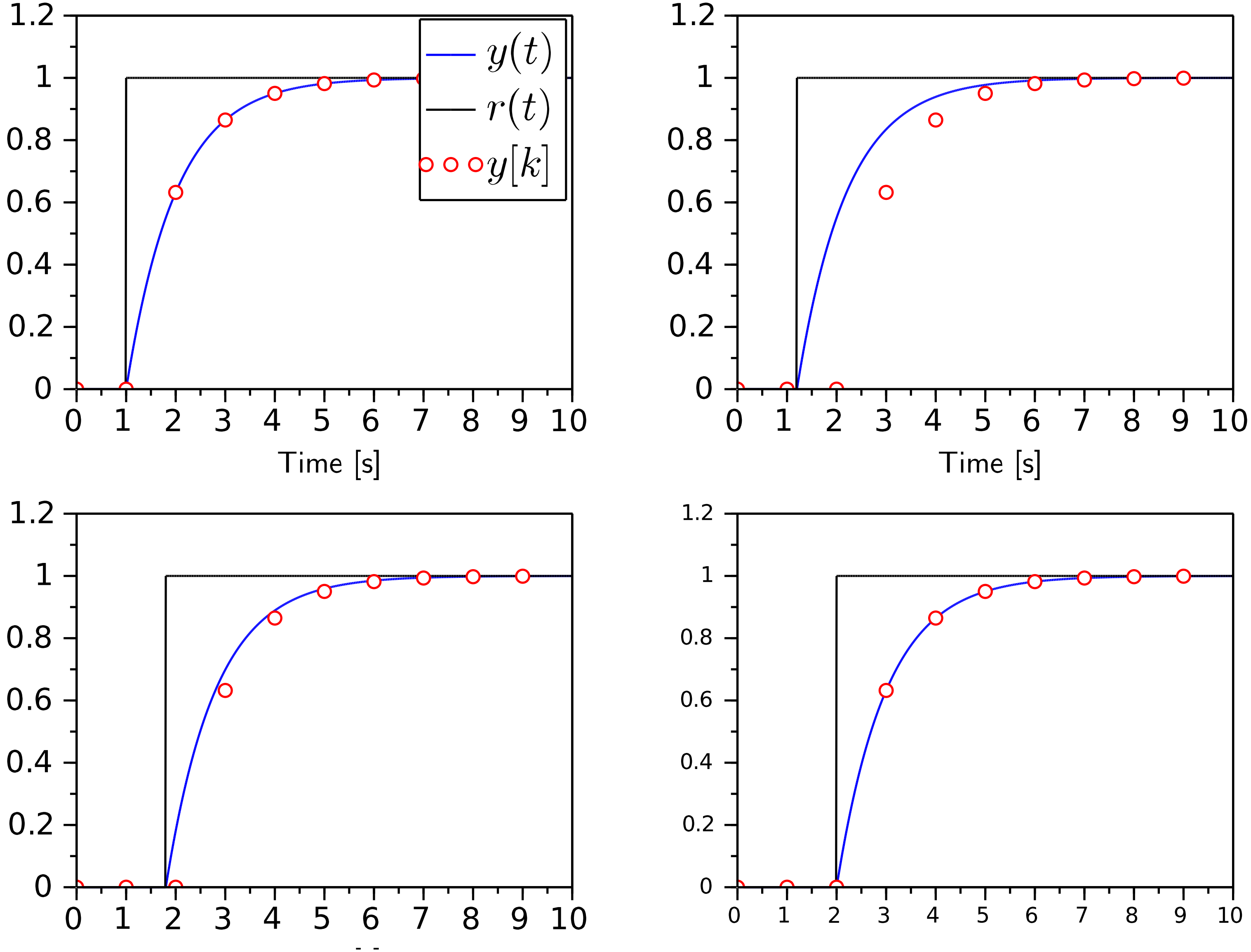
- The sampling rate should be chosen high enough.
- All signal components with frequencies higher than the Nyquist frequency must be removed before sampling. $\rightsquigarrow$ *Anti-aliasing filters*



Time dependence

- The presence of a clock makes computer-controlled systems time-varying.

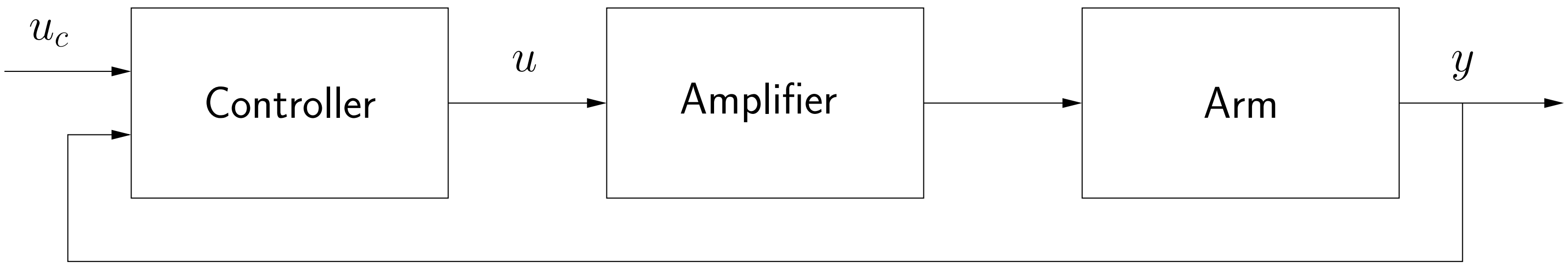



$y(t)$
$r(t)$
$y[k]$
Time [s]
Time [s]

A Naive Approach to Computer-Controlled Systems

- The computer controlled system behaves as a continuous-time system if the sampling period is sufficiently small!

Example: Controlling the arm of a disk drive



- Relation between arm position y and drive amplifier voltage u:

$$G(s) = \frac{c}{Js^2}$$

J - moment of inertia, c - a constant

- Simple servo controller (2DOF, lead-lag filter):

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

- Desired closed-loop polynomial with tuning parameter ω_0:

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

- Can be obtained with $a = 2\omega_0, \quad b = \omega_0/2, \quad K = 2\frac{J\omega_0^2}{c}$

Reformulation of the controller:

$$
\begin{aligned}
U(s) &= \frac{bK}{a} U_c(s) + KY(s) + K\frac{(a-b)}{(s+a)} Y(s) \\
&= K\left(\frac{a}{b} U_c(s) - Y(s) + X(s)\right)
\end{aligned}
$$

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

Euler method (approximating the derivative with a difference):

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

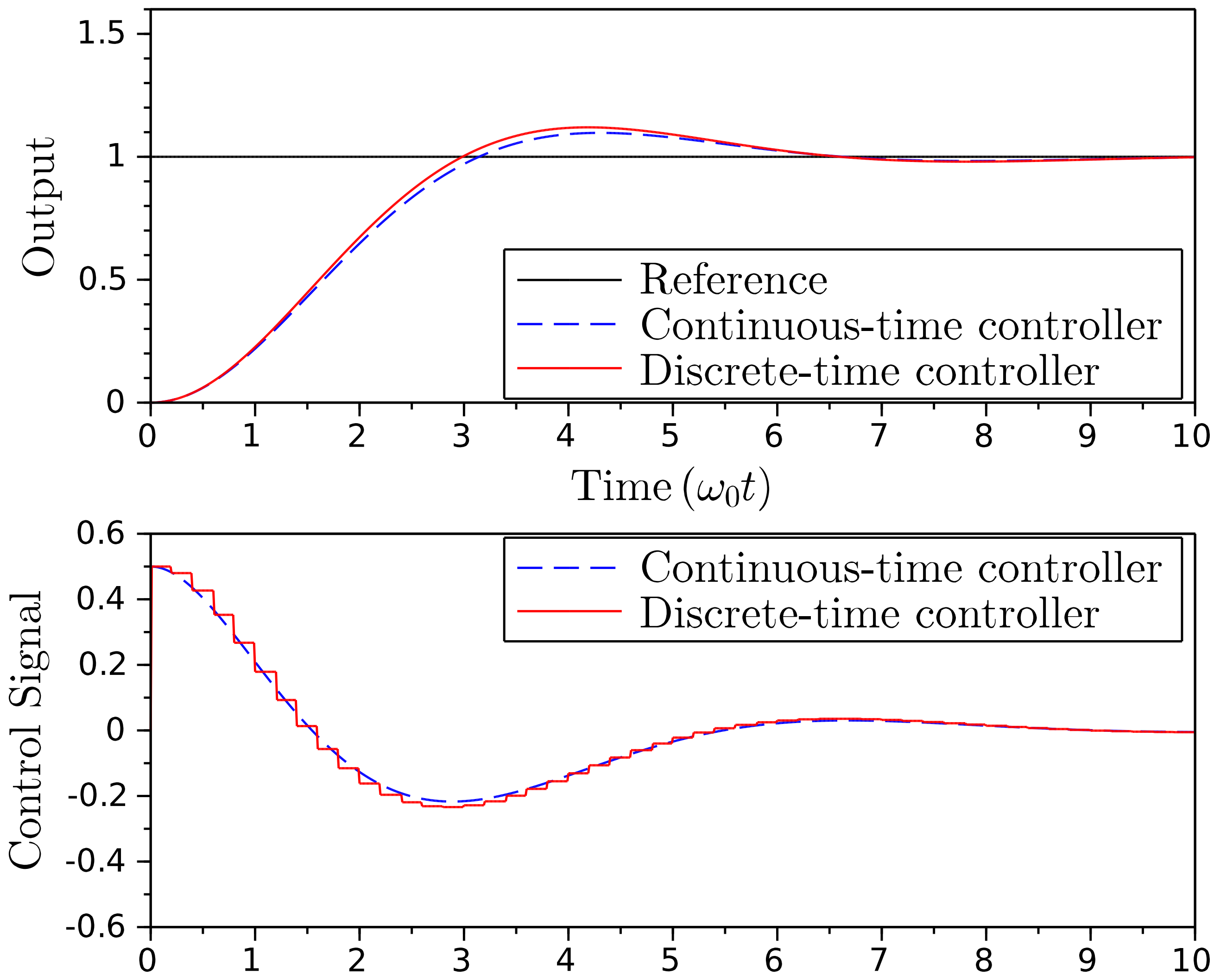
The following approximation of the continuous control law is then obtained:

$$\begin{aligned} u[k] &= K\left(\frac{b}{a}u_c[k] - y[k] + x[k]\right) \\ x[k+1] &= x[k] + \Delta((a-b)y[k] - ax[k]) \end{aligned}$$

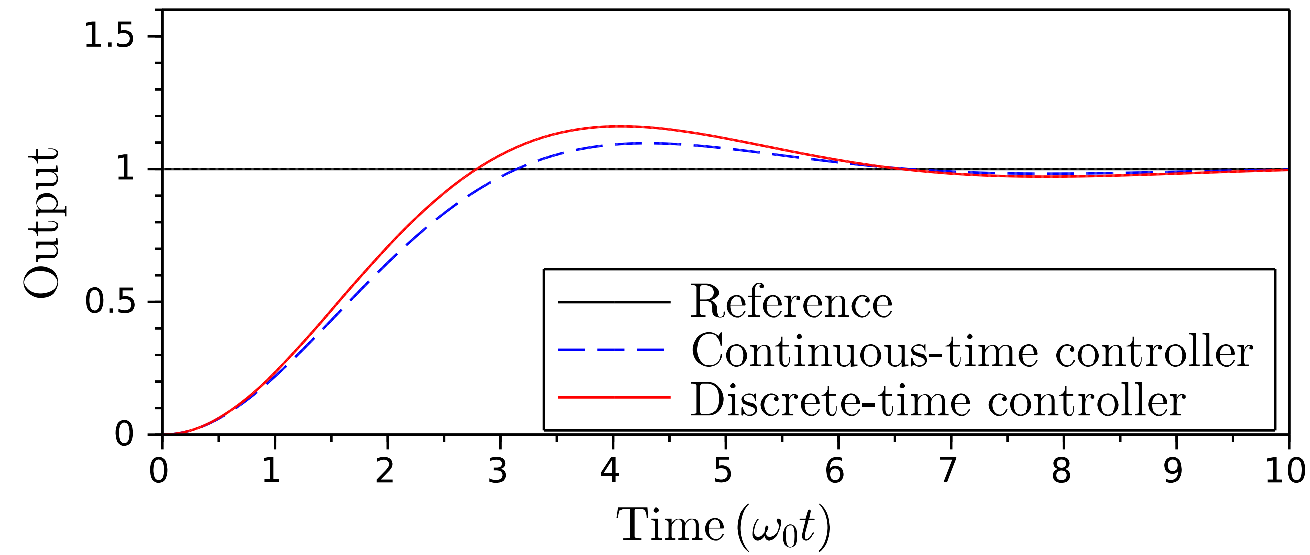
Computer program periodically triggered by clock:

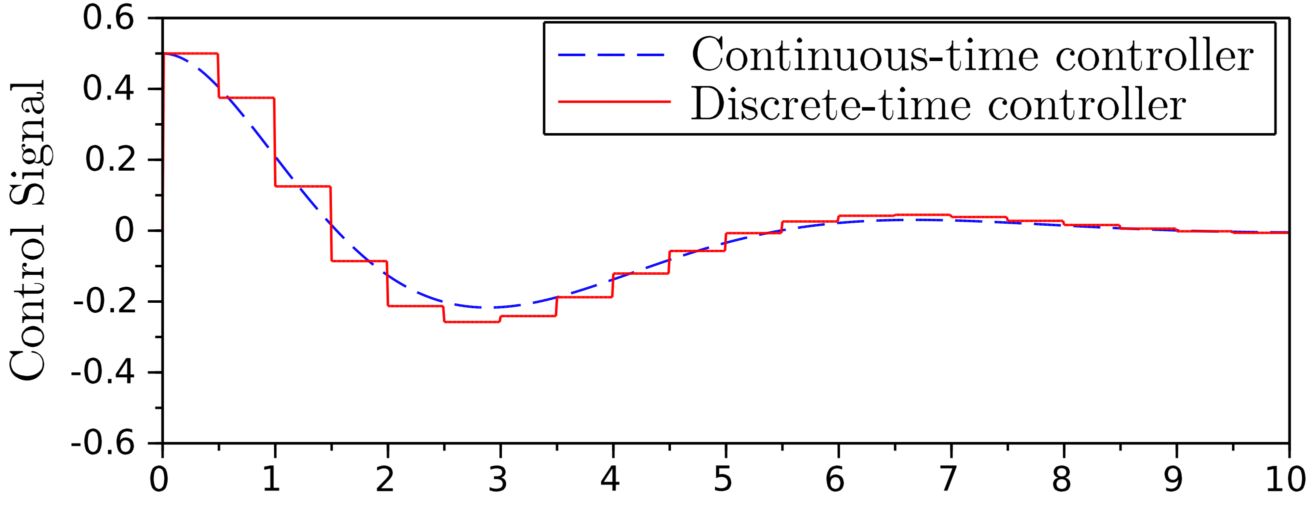
```
y: = adin(in1) {read process value}
u: = K*(a/b*us-y+x);
daout(u);      {output control signal}
newx: = x+Delta*((b-a)*y-a*x)
```

$$\Delta = 0.2/\omega_0$$

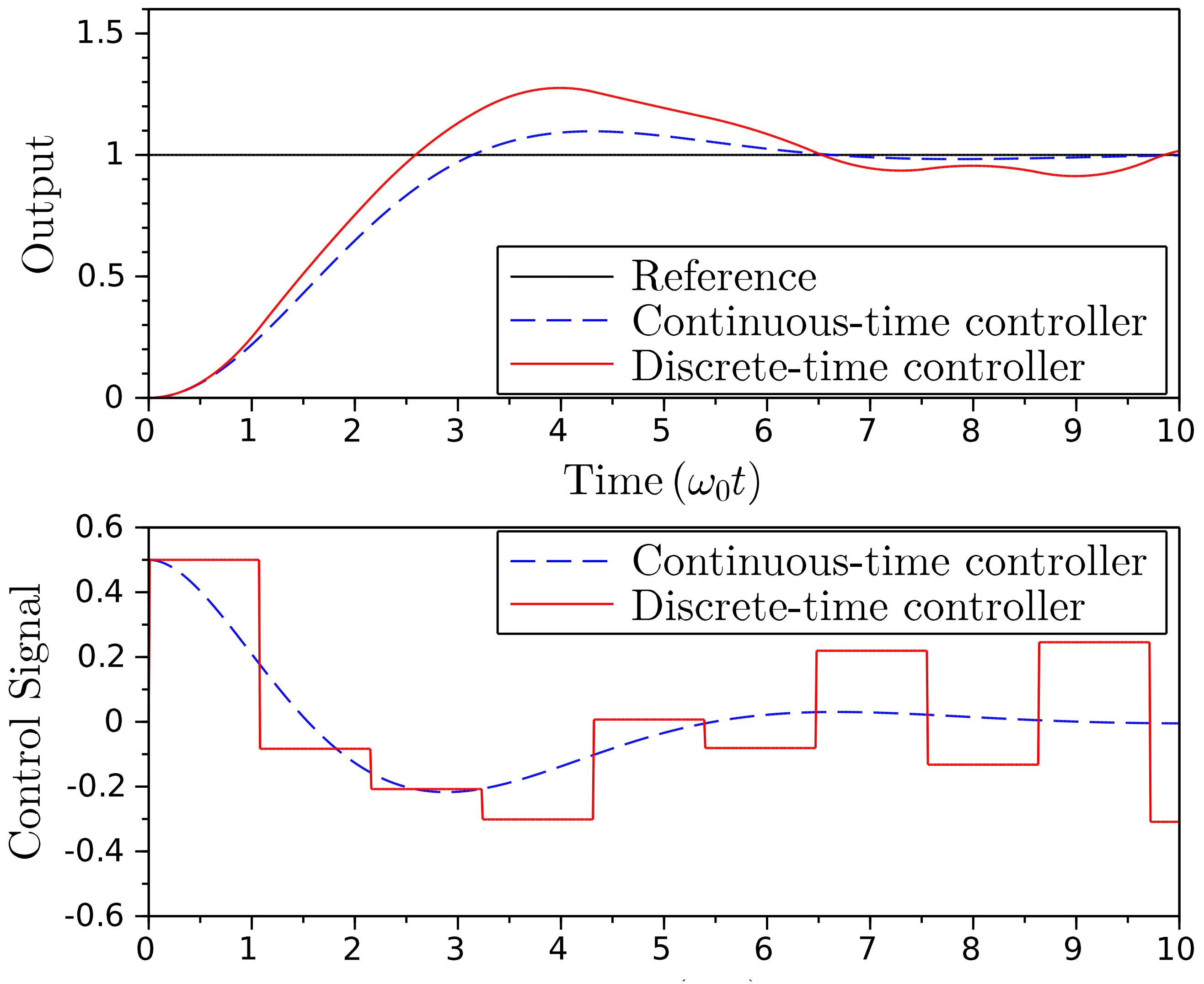


$$\Delta = 0.5/\omega_0$$





$$\Delta = 1.08/\omega_0$$



Deadbeat control

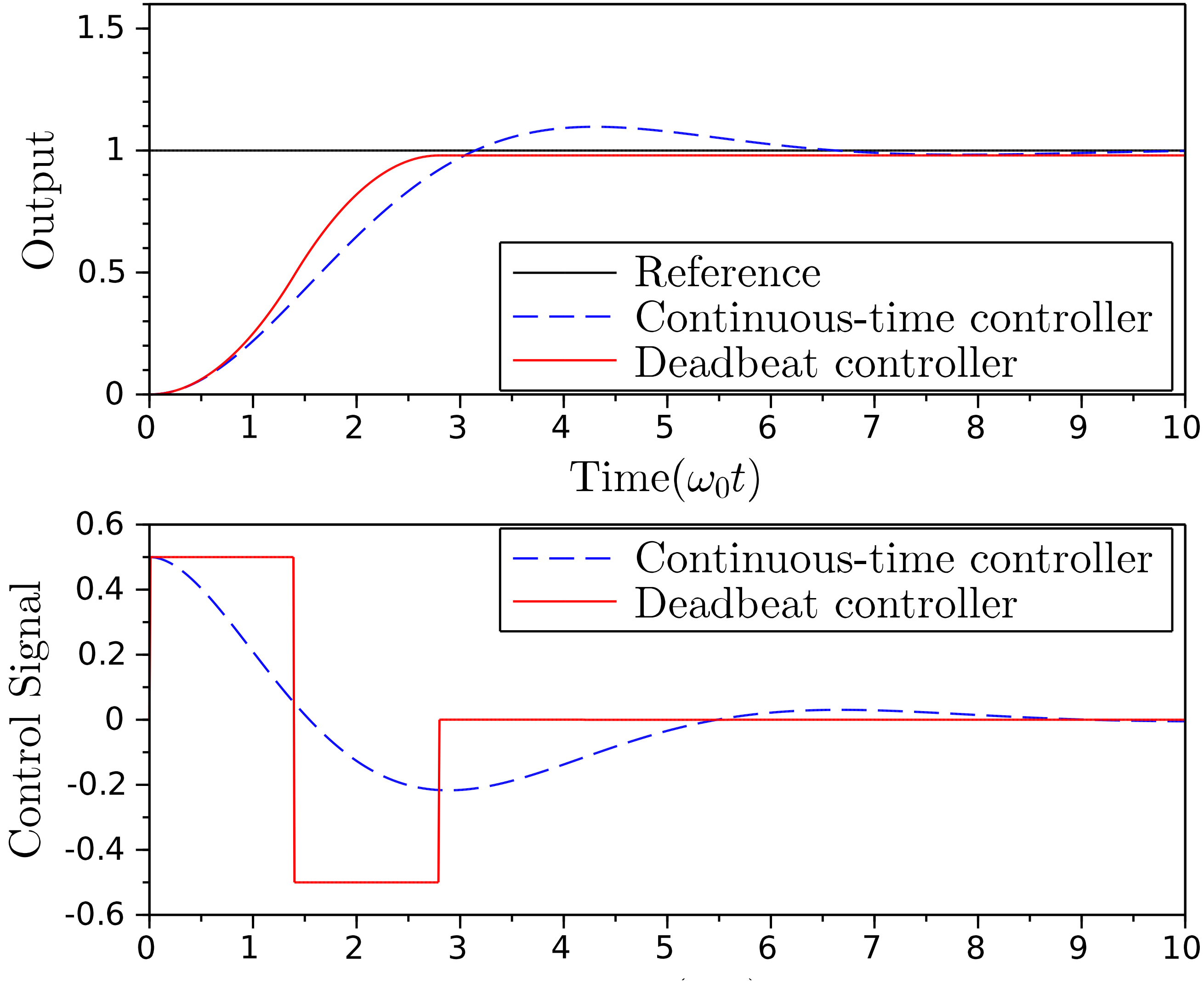
- The previous example seemed to indicate that a computer-controlled system will be inferior to a continuous-time example.
- This is not the case: The direct design of a discrete time controller based on a discretised plant offers control strategies with superior performance!
- Consider this controller structure

$$u[k] = t_0 u_c[k] - s_0 y[k] - s_1 y[k-1] - r_1 u[k-1]$$

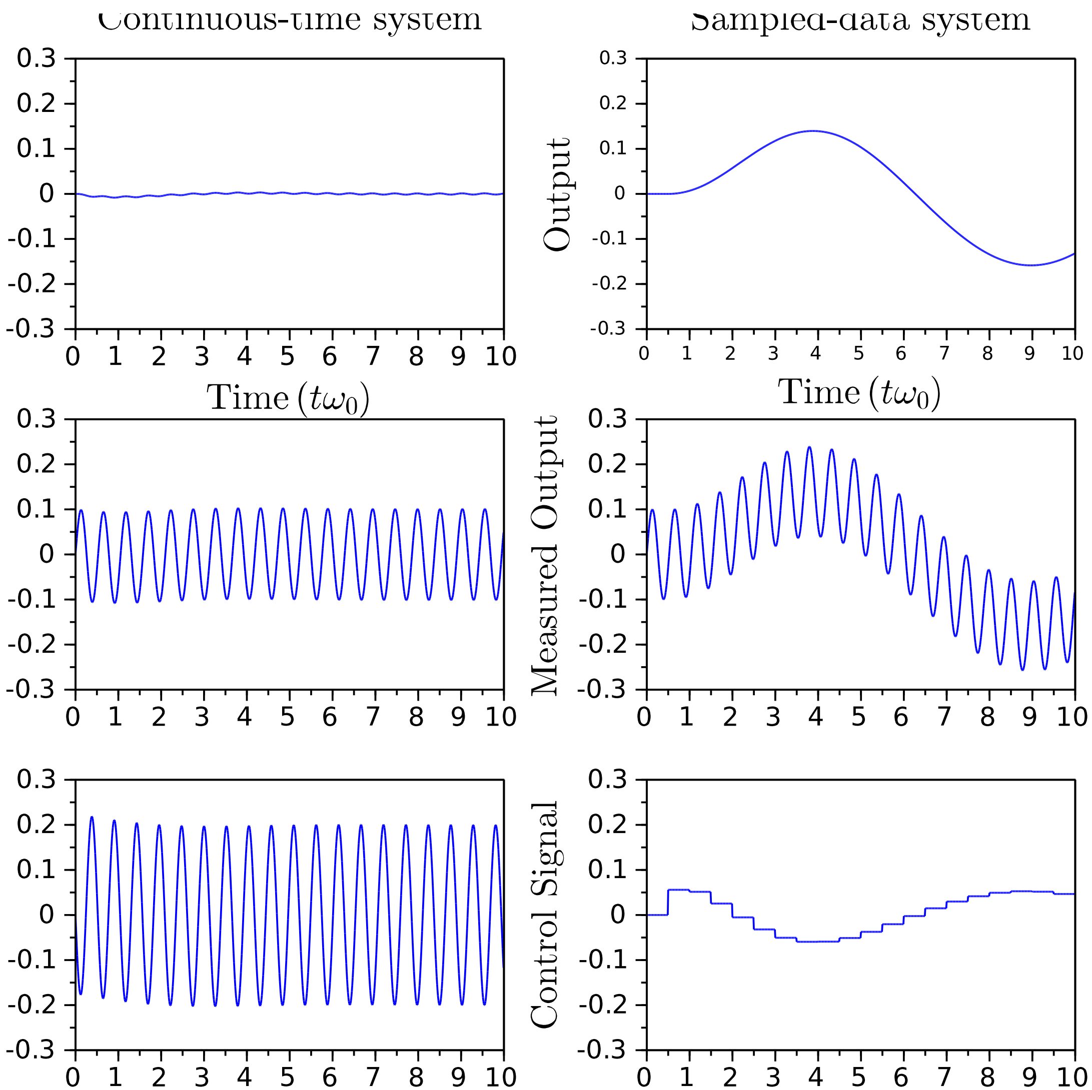
  with the long sampling period $\Delta = 1.4/\omega_0$.

- Sampling can initiated when the command signal is changed to avoid extra time delays due to the lack of synchronisation.

Deadbeat control



Anti-aliasing revisited - disk arm example



Sinusoidal measurement 'noise': $n = 0.1\sin(12t)$, $\omega_0 = 1, \Delta = 0.5$

Difference Equations

- The behaviour of computer-controlled systems can very easily described at the sampling instants by difference equations.
- Difference equations play the same role as differential equations for continuous-time systems.

Example: Design of the deadbeat controller for the disk arm servo system

- The disk arm dynamics with a control signal, that is constant over the sampling intervals, can be exactly described at sampling instants by

$$y[k] - 2y[k-1] + y[k-2] = \frac{c\Delta^2}{2J}(u[k-1] + u[k-2]). \tag{1}$$

- The Closed-loop system thus can be described by the equations

$$\begin{aligned} y[k] - 2y[k-1] + y[k-2] &= \frac{c\Delta^2}{2J}(u[k-1] + u[k-2]) \\ u[k] + r_1 u[k-1] &= t_0 u_c[k] - s_0 y[k] - s_1 y[k-1] \end{aligned}$$

- Eliminating the control signal (e.g. by using the shift-operator and $\alpha = \dfrac{c\Delta^2}{2J}$) yields:

$$
\begin{aligned}
y[k] + (r_1 - 2 + \alpha s_0)y[k-1] + (1 - 2r_1 + \alpha(s_0 + s_1))y[k-2] + (r_1 + \alpha s_1)y[k-3] \\
= \frac{\alpha t_0}{2}(u_c[k-1] + u_c[k-2])
\end{aligned}
$$

- The desired deadbeat behaviour

$$y[k] = \frac{1}{2}(u_c[k-1] + u_c[k-2])$$

can be obtained by choosing

$$r_1 = 0.75, \quad s_0 = 1.25/\alpha, \quad s_1 = -0.75/\alpha, \quad t_0 = 1/(4\alpha).$$

Is there a need for a theory for computer-controlled systems?

Examples have shown:

- Control schemes are possible that cannot be obtained by continuous-time systems.
- Sampling can create phenomena that are not found in linear time-invariant systems.
- Selection of sampling rate is important and the use of anti-aliasing filters is necessary.

These points indicate the need for a theory for computer controlled systems.

Inherently Sampled Systems

- Sampling due to the measurement
  - Radar
  - Analytical instruments (Glucose Clamps)
  - Economic systems
- Sampling due to pulsed operation
  - Biological systems