Signalbehandling for computer-ingeniører

COMTEK-5, E22

&

Signalbehandling

EIT-5, E22

6. Digital IIR Filters, The Impulse Invariant Method

Assoc. Prof. Peter Koch, AAU

The EIT-5 Study Regulation

Formål:
Analyse og filtrering af signaler er en disciplin, der er en forudsætning for alle specialiseringer i elektroniske systemer. Disciplinen anvendes indenfor automation, kommunikation, multimedie systemer, m.m. Kursets formål er at understøtte den studerende i at forstå centrale begreber, teorier og metoder til analyse og filtrering af analoge og digitale signaler, samt anvende teorier og metoder til analyse og filtrering af analoge og digitale signaler.

Læringsmål:

Yellow is identical to COMTEK-5

Viden
- Skal have viden om teorier og metoder til analyse og behandling af signaler på en computer
- Skal have viden om teorier og metoder til spektralestimering
- Skal have viden om teorier og metoder til design af analoge og digitale filtre (IIR/FIR)
- Skal have viden om teorierne og metodernes begrænsninger
- Skal have viden om sammenhæng mellem analyse af signaler i tids- og frekvensdomænet
- Skal have viden om teorier og metoder til transformation mellem forskellige domæner

Færdigheder
- Skal kunne anvende værktøjer til analyse, design og simulering af analoge og digitale signalbehandlingssystemer
- Skal kunne anvende teorier og metoder til spektralestimering herunder DFT/FFT
- Skal kunne demonstrere sammenhæng mellem frekvensopløsning, vinduesfunktioner og zero-padding
- Skal kunne anvende teorier og metoder til design af analoge og digitale filtre
- Skal kunne implementere IIR filtre vha. af bl.a. bilineær transformation og impuls invariant metoderne
- Skal kunne redegøre for betydningen af faselinearitet og gruppeløbstid
- Skal kunne kunne designe FIR filtre vha. vinduesmetoden
- Skal kunne redegøre for sammenhæng mellem filtres pol-/nulpunktsdiagrammer og frekvensrespons
- Skal kunne implementere filtre i praksis og herunder kunne gøre brug af hensigtsmæssig filterstruktur, kvantisering og skalering.

Kompetencer
- Skal kunne diskutere grundlæggende teorier og metoder til analyse og behandling af analoge og digitale signaler under anvendelse af korrekt terminologi
- Skal kunne vurdere muligheder og begrænsninger i forbindelse med teoriernes og metodernes anvendelse i praksis

AALBORG UNIVERSITY
DENMARK

EIT students have previously followed a course in **Beregningsteknik indenfor elektronikområdet 2 (BIE-2)**

Essential topics from that specific course…

- Tids-diskrete signaler og systemer
- Lineær tids-invariante systemer (LTI-systemer)
- Kausalitetsforhold og foldningsoperationer i LTI-systemer
- z-transformation
- z-transformeredes konvergensregioner og egenskaber
- Den inverse z-transformation
- Beregning og anvendelse af den inverse z-transformation
- Lineære differensligninger med konstante koefficienter
- Stabilitets- og kausalitetsforhold
- Repræsentation af tids-diskrete signaler og systemer i frekvensdomænet
- Nyquist-Shannon's samplingssætning

Preliminary course overview – the final 2/3

6. Digital IIR filters, Impulse Invariant Method
7. Digitale IIR filters, IIM again, The Bilinear Transform
8. Digitale FIR filters, The Window Method
9. Freq. Transformations and Freq. Analysis
10. Implementation of Digital Filters, numerical aspects
11. Discrete Fourier Transform
12. Discrete Fourier Transform, cont.
13. Fast Fourier Transform
14. Short Time Fourier Transform

AALBORG UNIVERSITY
DENMARK

What is signal processing..?

”Signal Processing” is a term, rooted in mathematical theories and methods, which expresses all the things we could possibly perform on a signal – modification, analysis and transport/storage.

Signal processing actually has been around since the early days of electronics, i.e., more than 100 years – what is done with analog circuits can be considered as signal processing, typically filtering…

However, it is not until 1948 that ”signal processing” as an individual term becomes widely known.

This year Claude Shannon publishes his very famous paper ”A Mathematical Theory of Communication” in Bell Systems Technical Journal. The paper discusses the term ”entropy”. One year later, the paper is re-published, but now; A -> The

Also in 1948 the American researcher John Tukey develops some of the first methods for discrete-time spectral estimation – a concept which is still heavily used today (… and we will address these methods in the second half of the course).

What is signal processing..??

Further, also in 1948 the American research Richard W. Hamming develops the first digital (binary) error-correcting codes.

BTW, 1948 is also the year where William Bradford Shockley and his research team develops the transistor at Bell Telephone Laboratory.

Beside, it should be mentioned that 1948 is also the year where the LP record is introduced; Peter Goldmark, head of CBS Laboratories, publishes the paper ”The Columbia Long-Playing Microgroove Recording System”. The LP record lasted for two generarions…

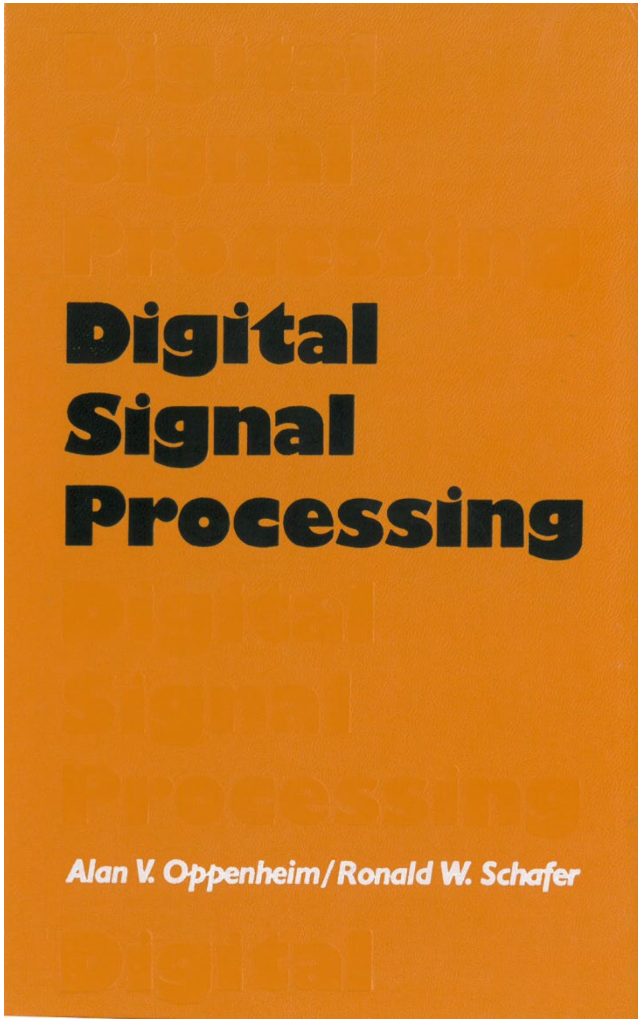
In the signal processing community, 1948 is therefore know as the ”Annus Mirabilis”…

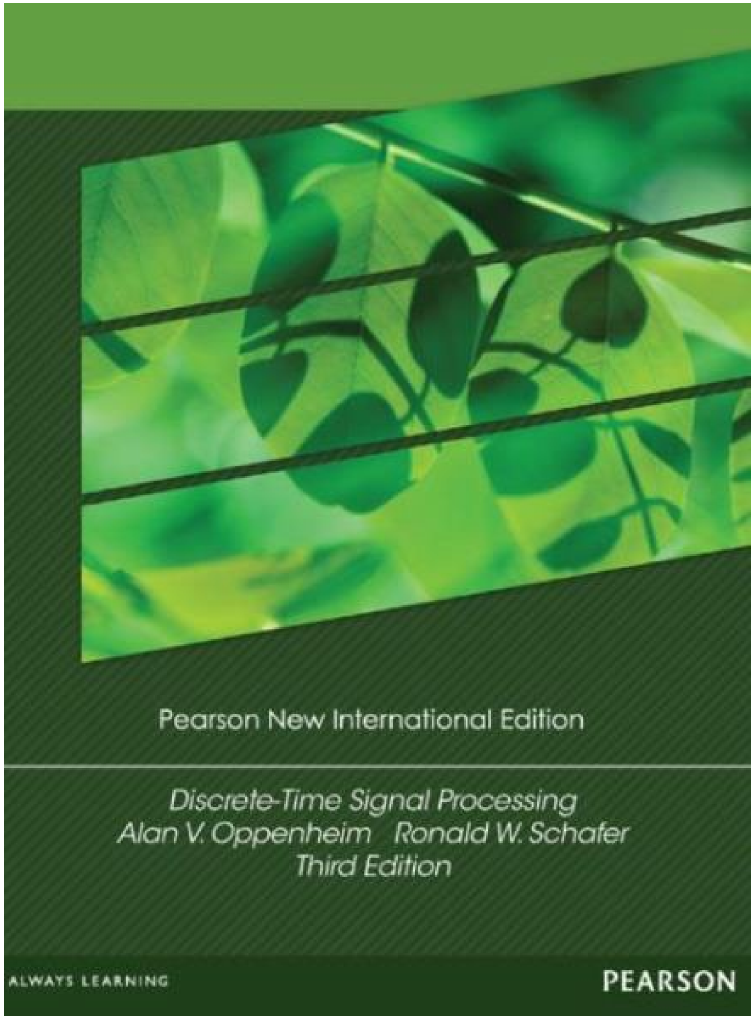
During the 50's, 60's and 70's many mathematical theories and methods which are fundamental to modern discrete-time (digital) signal processing are being developed.

In 1969, the first exhaustive text book on digital signal processing is published…

Oppenheim & Schafer publish their first book in 1975…

Later is was renamed to Discrete-Time Signal Processing and now it is in its 3rd edition (1989, 1999, 2014)





AALBORG UNIVERSITY
DENMARK

But, what is signal processing..??

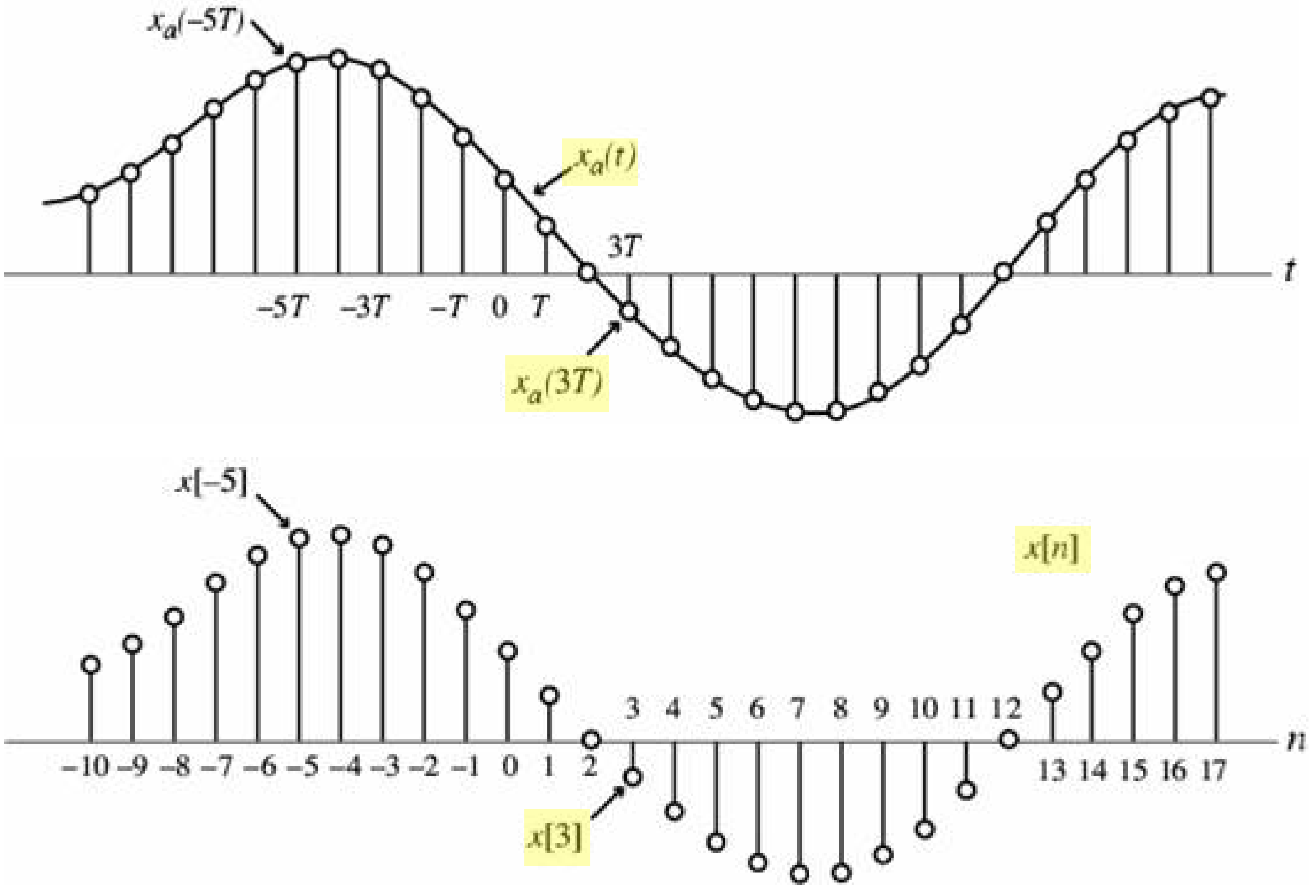
From the early 60’s we have seen a tremendous development in semiconductor technologies which has paved the way for continuously more and more powerful computers – a necessity in order to conduct digital processing of signals.

Essentially, signal processing (or digital signal processing, DSP) is a computer program which executes an algorithm that ”do something” to a signal which is applied as input to that program. Therefore, signal processing is often ”invissible” to the user…

Today, signal processing is ”embedded” in almost all devices/applications and we are all doing a lot of signal processing every single day – typically without thinking about it…

The organization Institute of Electrical and Electronics Engineers (IEEE) has produced a short video which tries to envision the way signal processing has penetrated our daily life…

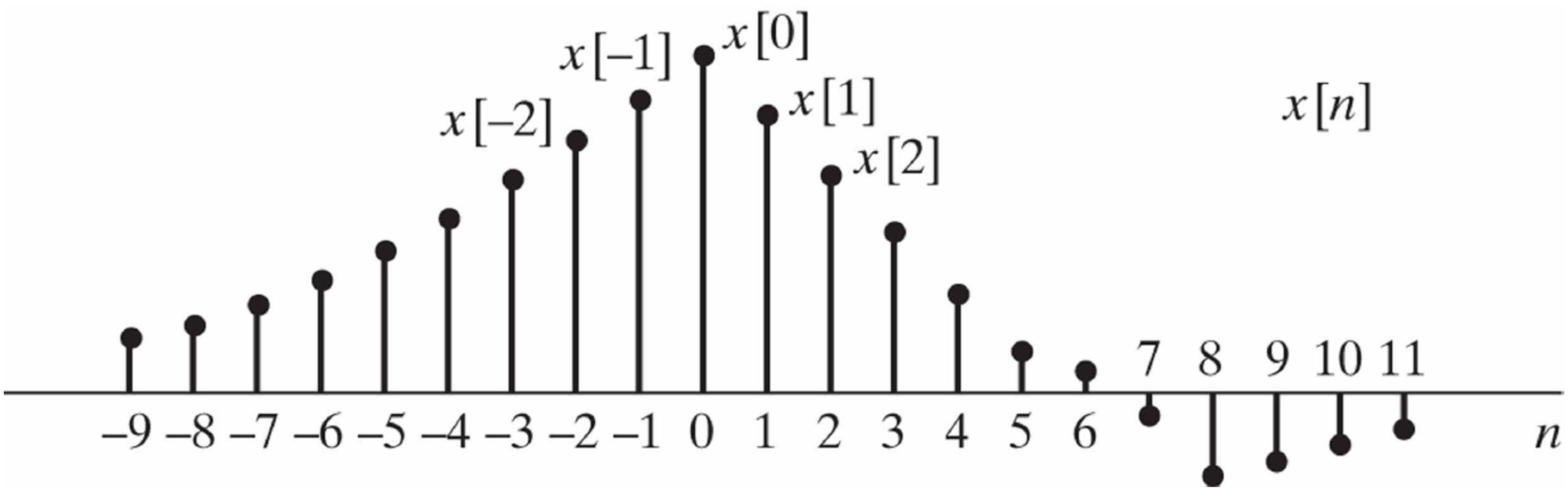
A continuous-time signal becomes a discrete-time signal…



When a signal is discrete in time, we refer to it as a sequence…

Talfølge: $x = \{x[\mathrm{n}]\} \quad \mathrm{n} \in Z$

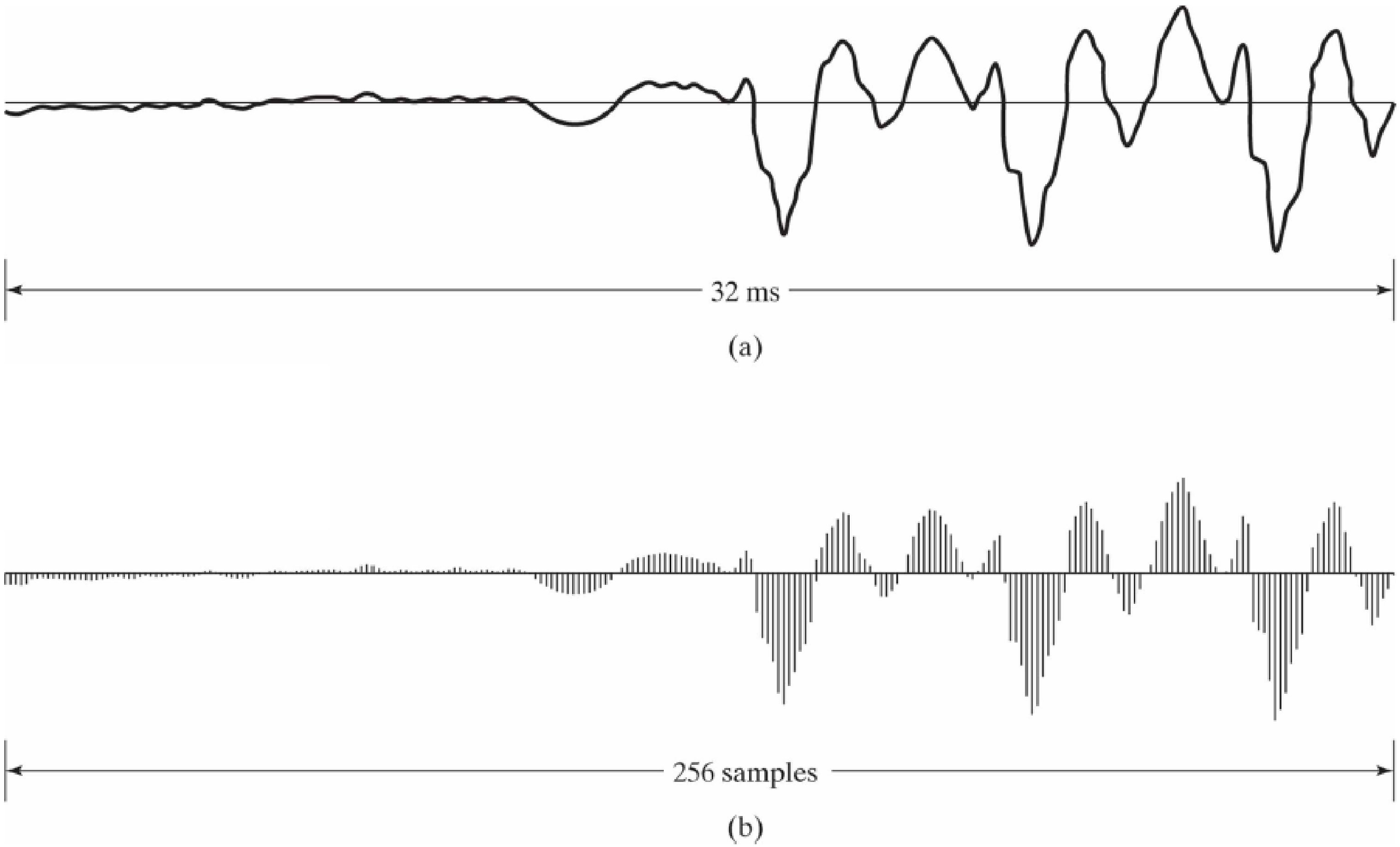
$$\cdots, x[-1], x[0], x[1], \cdots$$



$x[n]$ is defined only for the values n.

– thus, $x[n]$ is NOT zero in between the individual samples..!

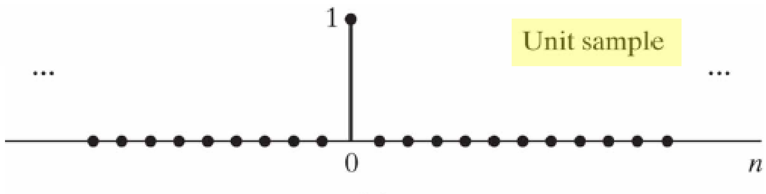
Sample interval and **sample frequency** are reciprocal – what is the sample frequency in this example..??



AALBORG UNIVERSITY
DENMARK

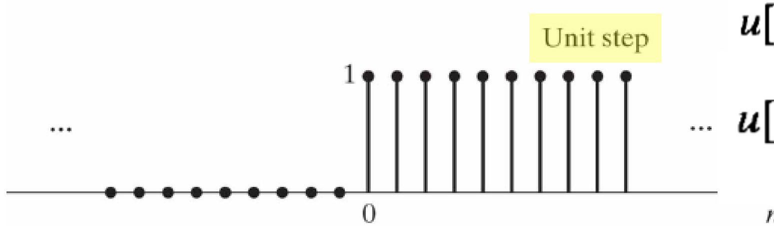
Important sequences…

$$\delta[n] = \begin{cases} 0, & n \neq 0, \\ 1, & n = 0. \end{cases}$$



(a)

$$u[n] = \begin{cases} 1, & n \geq 0, \\ 0, & n < 0. \end{cases}$$

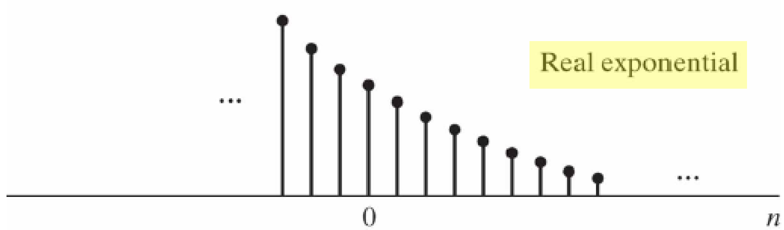


(b)

$$u[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \cdots$$

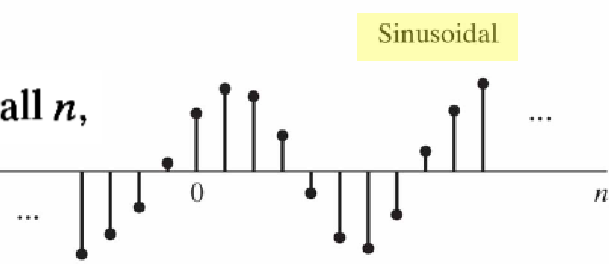
$$u[n] = \sum_{k=0}^{\infty} \delta[n-k].$$

$$x[n] = A\alpha^n.$$



(c)

$$x[n] = A\cos(\omega_0 n + \phi), \qquad \text{for all } n,$$



(d)

Discrete-Time Filters

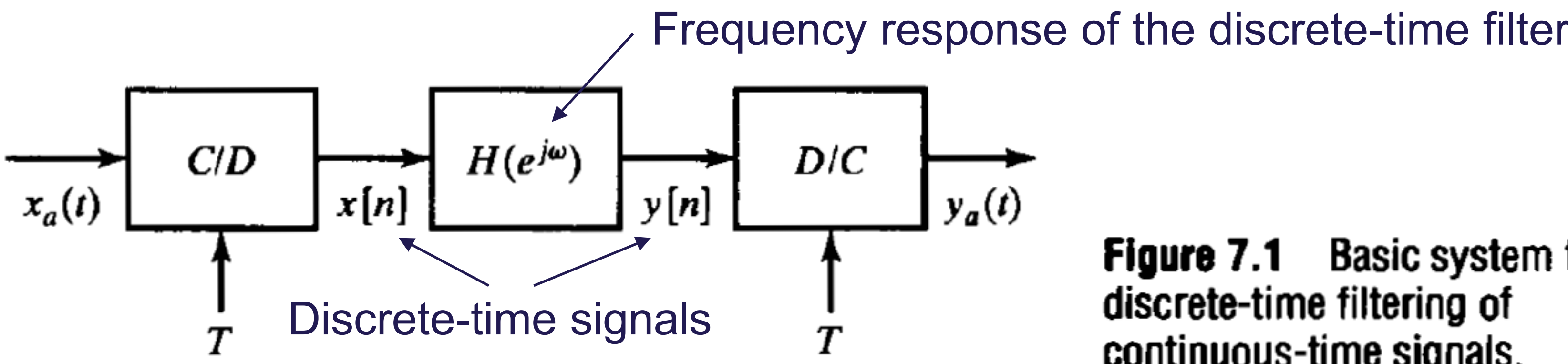


Figure 7.1 Basic system for discrete-time filtering of continuous-time signals.

Relation between continuous-time and discrete-time frequency; $\omega = \Omega T = 2\pi f T = \frac{2\pi f}{f_s}$

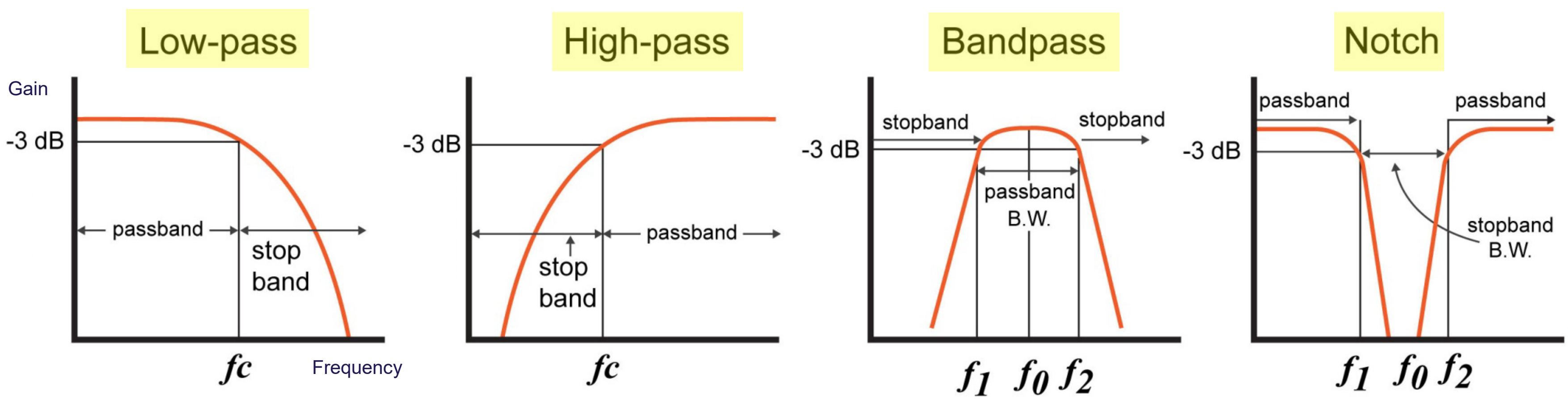
From this expression, we clearly see that 2π corresponds to the sample frequency

Specification of the effective filter; $H_{\text{eff}}(j\Omega) = \begin{cases} H(e^{j\Omega T}), & |\Omega| < \pi/T, \\ 0, & |\Omega| > \pi/T. \end{cases}$

Specification of the discrete-time filter; $H(e^{j\omega}) = H_{\text{eff}}\left(j\frac{\omega}{T}\right), \qquad |\omega| < \pi.$

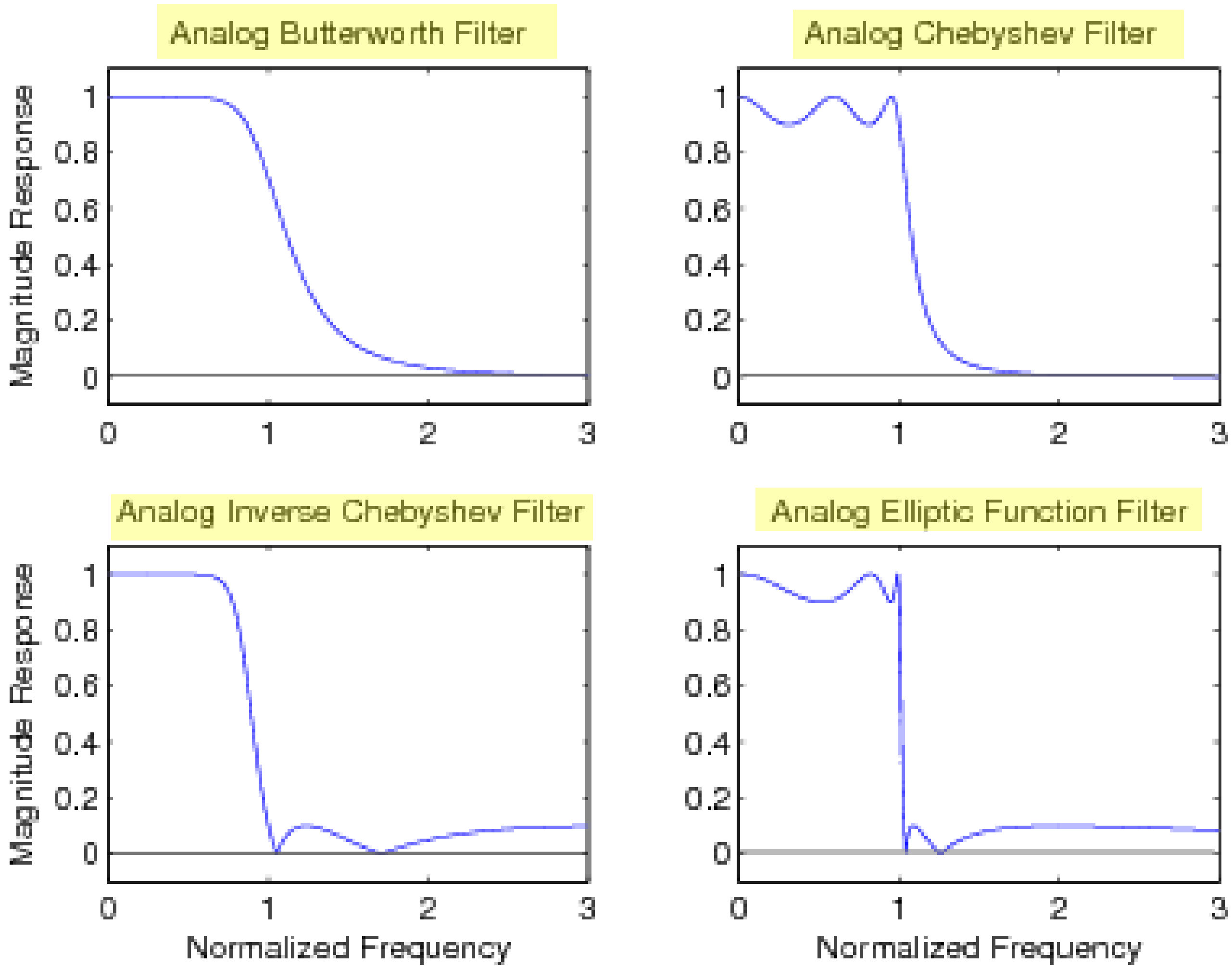
AALBORG UNIVERSITY
DENMARK

Filters – different functional types



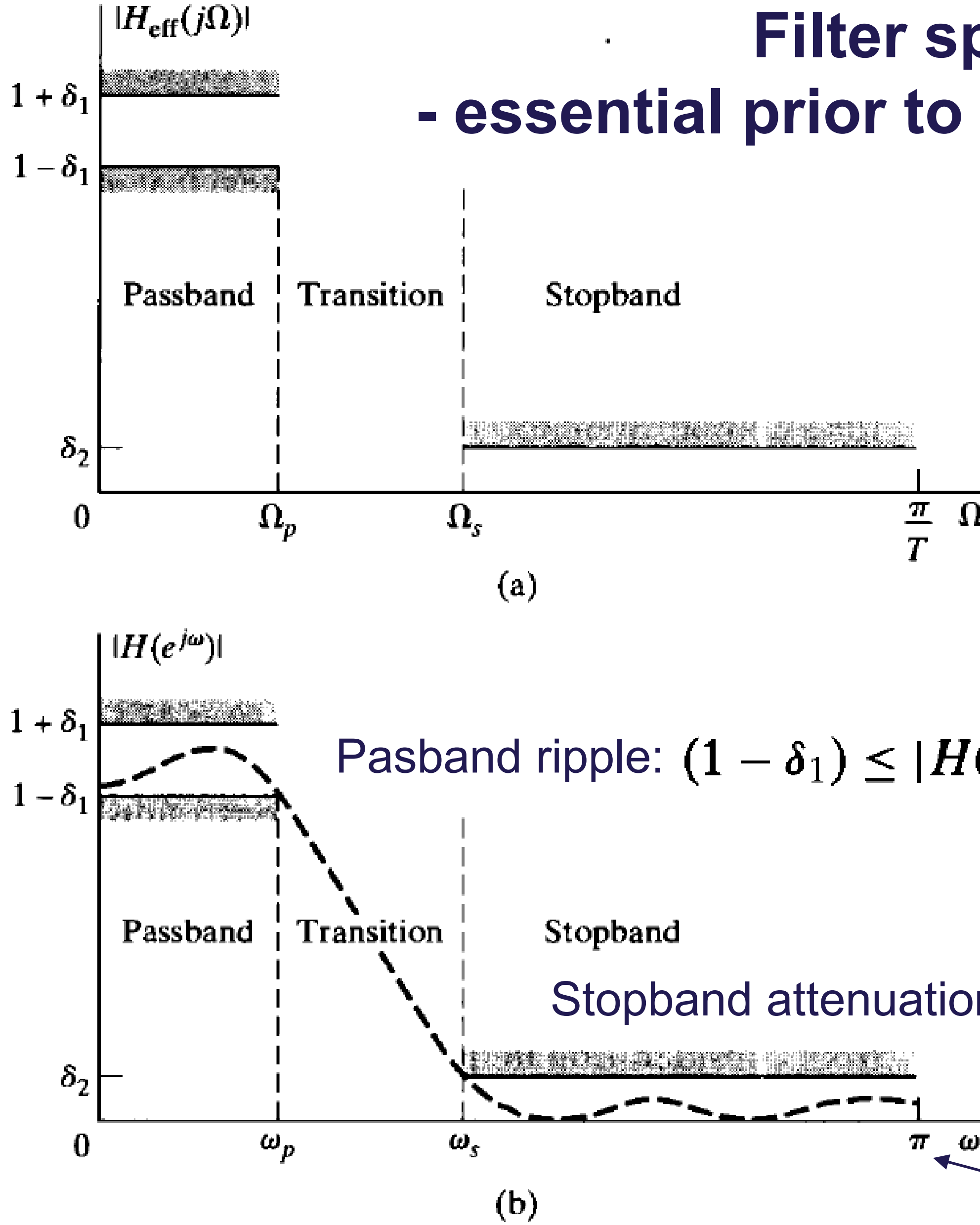
AALBORG UNIVERSITY
DENMARK

Frequency selective filters – various matematical models



AALBORG UNIVERSITY
DENMARK

Filter specification
- essential prior to the design of our filter



Pasband ripple: $(1-\delta_1) \le |H(e^{j\omega})| \le (1+\delta_1), \qquad |\omega| \le \omega_p$

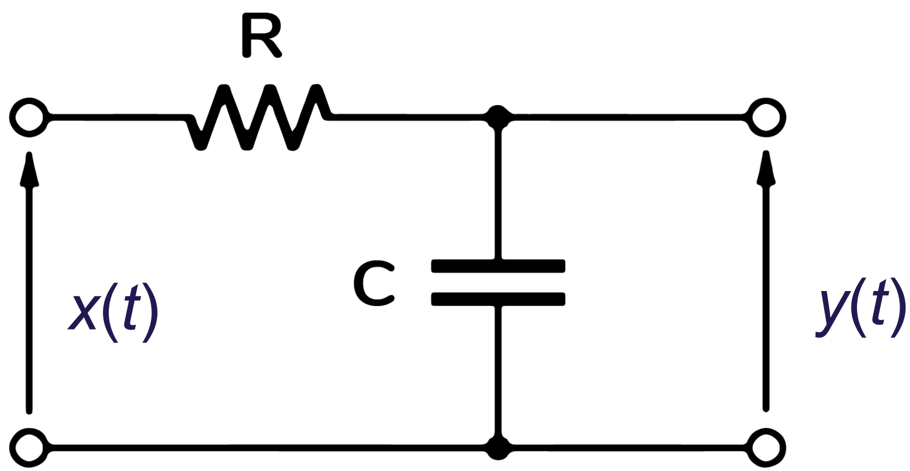
Stopband attenuation: $|H(e^{j\omega})| \le \delta_2, \qquad \omega_s \le |\omega| \le \pi$

In the discrete-time domain, everything has be related to the sample freqency

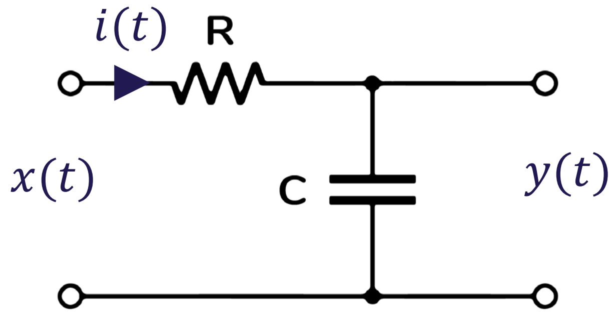
Figure 7.2 (a) Specifications for effective frequency response of overall system in Figure 7.1 for the case of a lowpass filter. (b) Corresponding specifications for the discrete-time system in Figure 7.1.

So, let's see if we can design a digital filter…

Basically what we will do is to apply ALL the math that we can possibly think of on a very simple circuit – an RC circuit with one restistor and one capacitor.



We will establish input/output relations in various domains and show how it all fits nicely together…



Using Kirchhoff Voltage Law, we can write that $x(t) = V_R(t) + V_C(t) = V_R(t) + y(t)$

Next, using Ohm's law we can state that $V_R(t) = R \cdot i(t)$, where $i(t)$ is the current in the mesh, i.e,

$$x(t) = R \cdot i(t) + y(t)$$

In this expression we have *i*(*t*) as an unknown variable, which however can be eliminated by using the voltage/current relation for a capacitor;

$$i(t) = C \cdot \frac{d}{dt} y(t)$$

and thus

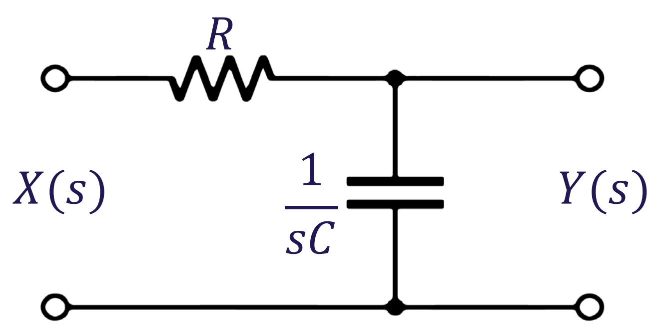
$$\boxed{x(t) = RC \cdot \frac{d}{dt} y(t) + y(t)}$$

$$x(t) = RC \cdot \frac{d}{dt} y(t) + y(t)$$

This equation represents an I/O relation for the filter in the time domain.

OBS…! The little circuit, consisting of one resistor and one capacitor, is continuously solving this 1'st order ordinary differential equation…!!

Now, let's do Laplace transform of the circuit…



Using the well-known voltage-divider equation, we can establish an expression for the output;

$$Y(s) = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} X(s)$$

$$Y(s) = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} X(s)$$

$$Y(s) = \frac{1}{sRC + 1} X(s)$$

$$\boxed{H(s) = \frac{Y(s)}{X(s)} = \frac{1}{sRC + 1}}$$

The transfer function *H*(*s*) is also an I/O relation for the system – but now in the *s*-domain…

…and it is easy to get back to the time domain using the inverse Laplace;

$$x(t) = \mathcal{L}^{-1}\{X(s)\}$$

$$x(t) = \mathcal{L}^{-1}\{sRC \cdot Y(s) + Y(s)\}$$

$$x(t) = RC \cdot \frac{d}{dt} y(t) + y(t) + y(0)$$

AALBORG UNIVERSITY
DENMARK

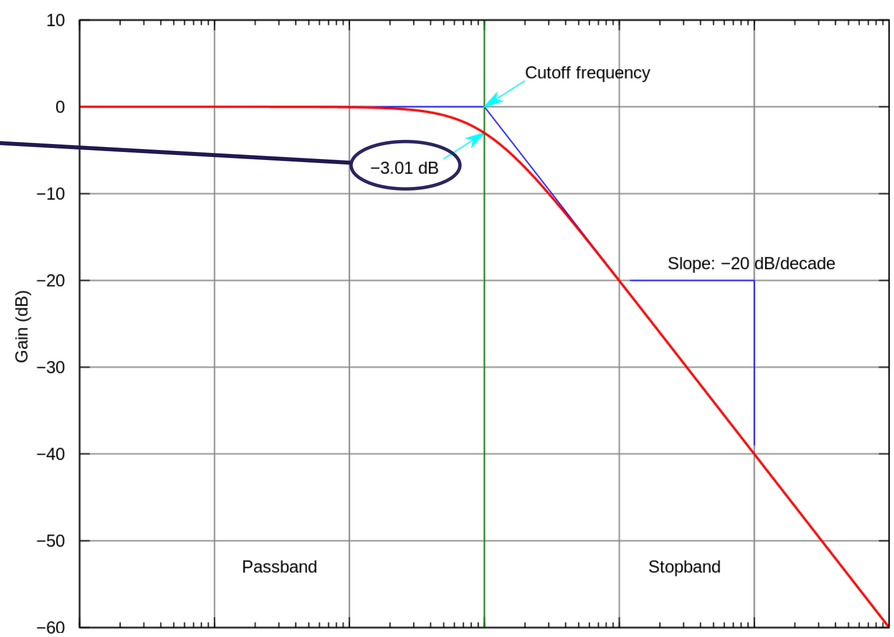
$$H(s) = \frac{Y(s)}{X(s)} = \frac{1}{sRC + 1} \qquad x(t) = RC \cdot \frac{d}{dt} y(t) + y(t) + y(0)$$

From the transfer function as well as from the differential equation we see that the parameter *RC* appears, and thus we may conclude that it impacts the overall working of the circuit. Let's look at the amplitude response, i.e., $s = j\Omega$ in $H(s)$;

$$|H(j\Omega)| = \left|\frac{1}{j\Omega RC + 1}\right| = 1 \quad \text{for} \quad \Omega = 0$$

$$= \frac{1}{\sqrt{2}} \quad \text{for} \quad \Omega = \frac{1}{RC}$$

$\frac{1}{\sqrt{2}}$

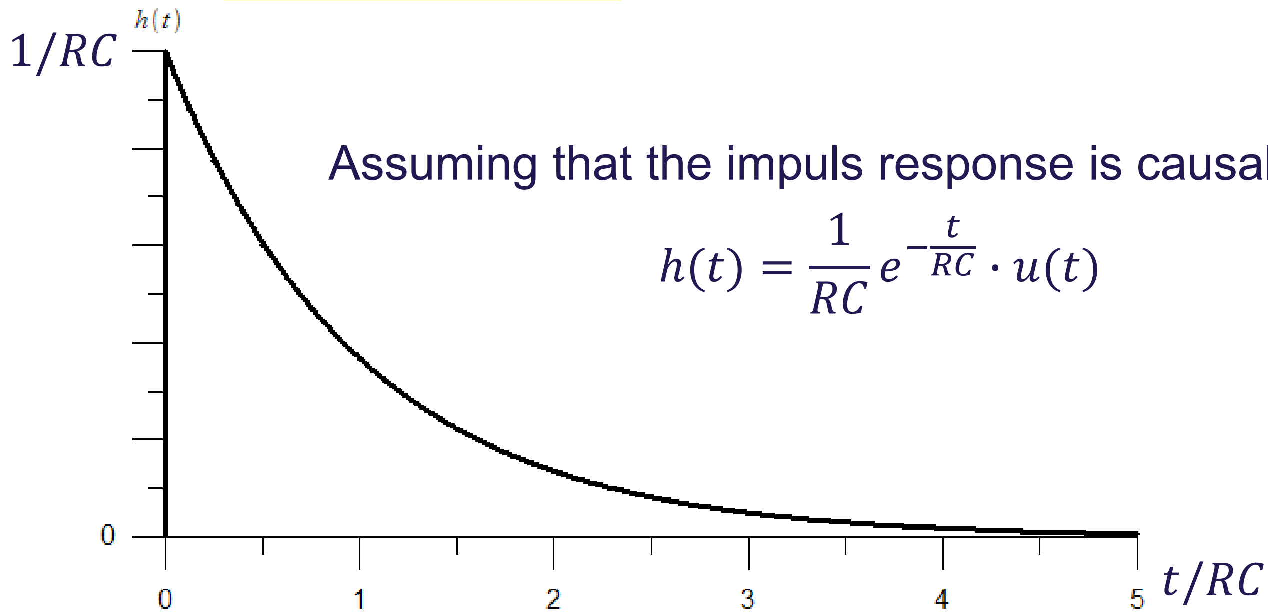


$\Omega = \frac{1}{RC}$

We may also do inverse Laplace of the transfer function $H(s)$;

$$
\begin{aligned}
h(t) &= \mathcal{L}^{-1}\{H(s)\} \\
&= \mathcal{L}^{-1}\{\frac{1}{sRC+1}\} \\
&= \mathcal{L}^{-1}\{\frac{1/RC}{s+1/RC}\} \\
&= \frac{1}{RC}e^{-\frac{t}{RC}}
\end{aligned}
$$

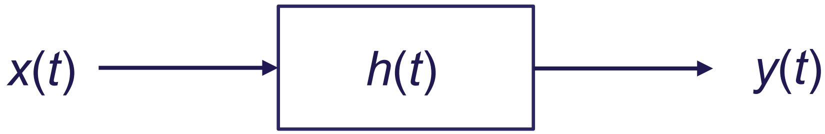
which is known as the impluse response.



Assuming that the impuls response is causal, we get;

$$h(t) = \frac{1}{RC}e^{-\frac{t}{RC}} \cdot u(t)$$

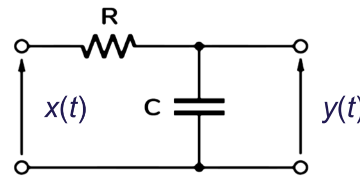
Now, given the impulse response, we can write the convolution integrale;



$$y(t) = h(t) * x(t)$$

$$= \int_{\tau=-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$$

Wow…! Think about it – this little circuit



constantly calculates the convolution between the input signal *x*(*t*) and the impulse response *h*(*t*)…

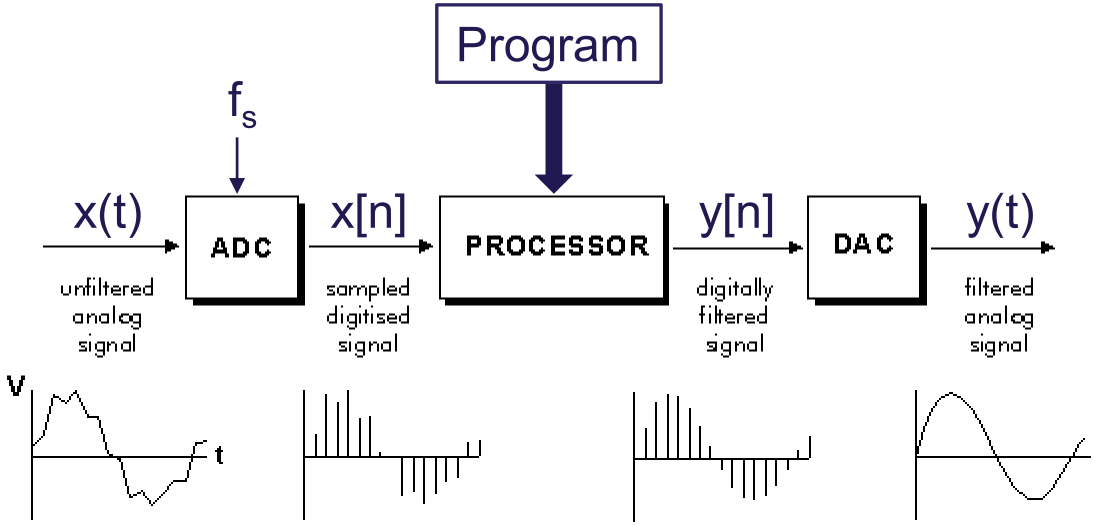
So, in conclusion we can state;

Given *H*(*s*) or equivalently *h*(*t*), then we have the totale I/O relation for the Linear Time-Invariant (LTI) system

Our purpose now is to consider the possibility for designing a ”digital” (or discrete-time) counterpart of the little analog RC-filter…

You may put it differently; Can we design a computer program which, seen from the input-to-output behaves just as the RC circuit..?

What is signal processing system...

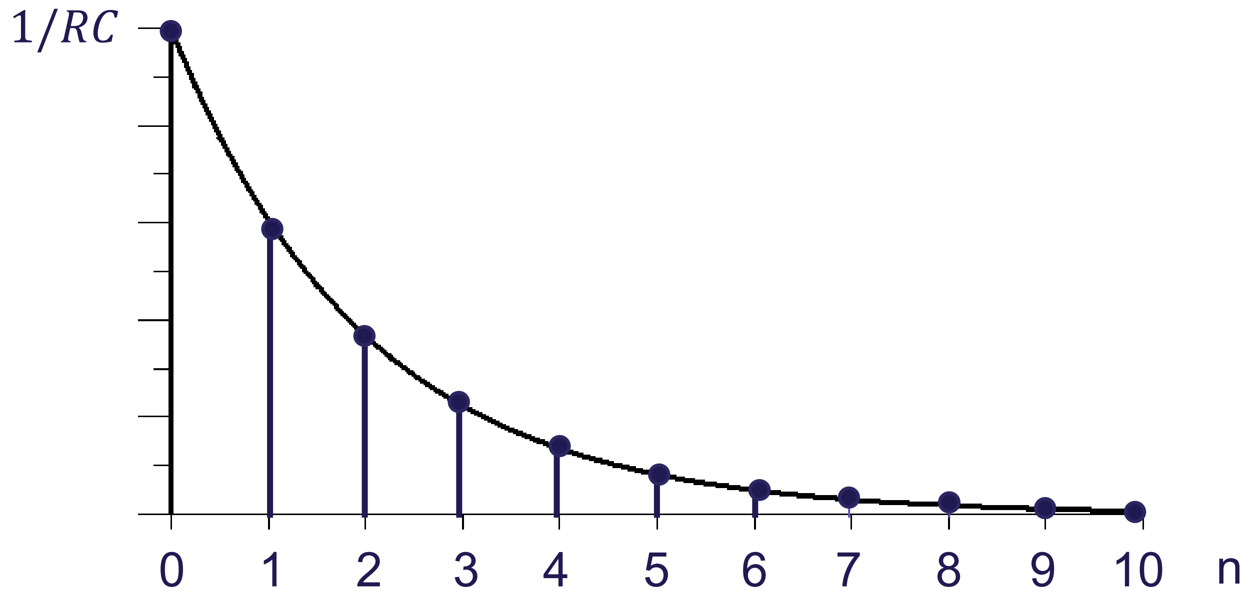


Our task therefore is to design a discrete-time algorithm, when executed on the processor, provides the same I/O relation between x(t) and y(t) as the RC circuit.

AALBORG UNIVERSITY
DENMARK

Now, since we have already concluded that the impulse response describes completely the systems, and since we know the impulse response for the analog filter, then an idea is to derive a discrete-time version of $h(t)$.

$$h(t) = \frac{1}{RC} e^{-\frac{t}{RC}} \cdot u(t)$$



$$h[nT] = h[n] = T \cdot h(t)|_{t=nT}$$ Equ. 2, p.522

where $T = 1/f_s$

So, now we can write an expression for the discrete-time impulse response

$$h[n] = \frac{T}{RC} e^{-\frac{nT}{RC}} \cdot u[n]$$

Or alternatively;

$$h[n] = \frac{T}{RC} e^{-\frac{nT}{RC}} \quad n \geq 0$$

Next, based on $h[n]$ we can express one possible I/O relation for the discrete-time system; the convolution sum

$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]$$

$$y[n] = h[n] * x[n]$$

$$= \sum_{k=-\infty}^{\infty} h[k] \cdot x[n-k]$$

$$= \frac{T}{RC} \sum_{k=-\infty}^{\infty} e^{\frac{-kT}{RC}} \cdot x[n-k]$$

Useful…???

Now that we have an expression for the discrete-time impulse response $h[n]$, we can similarly derive an expression for the transfer function $H(z)$;

$$H(z) = \mathcal{Z}\{h[n]\}$$

$$H(z) = \sum_{n=-\infty}^{\infty} h[n] \cdot z^{-n}$$

$$H(z) = \sum_{n=0}^{\infty} h[n] z^{-n}$$

$$H(z) = \sum_{n=0}^{\infty} \frac{T}{RC} e^{\frac{-nT}{RC}} \cdot z^{-n}$$

$$H(z) = \frac{T}{RC} \sum_{n=0}^{\infty} \{e^{\frac{-T}{RC}} \cdot z^{-1}\}^{n}$$

Next, we need to bring this expression onto a closed form…

AALBORG UNIVERSITY
DENMARK

$$H(z) = \frac{T}{RC} \sum_{n=0}^{\infty} \{e^{\frac{-T}{RC}} \cdot z^{-1}\}^n$$

We see that this is a geometric series (på dansk vil vi kalde det en uendelig kvotient-række), where the quotient $q = e^{\frac{-T}{RC}} \cdot z^{-1}$

$$H(z) = \frac{T}{RC} \cdot \frac{q^0 - q^{\infty+1}}{1-q} = \frac{T}{RC} \cdot \frac{1}{1 - e^{\frac{-T}{RC}} \cdot z^{-1}} \quad ROC: |z| > e^{\frac{-T}{RC}}$$

…and thus we can simplify this expression for the discrete-time transfer function;

$$\mathrm{H(z)} = \frac{b}{1 - az^{-1}} \quad \text{where} \quad b = \frac{T}{RC} \quad \text{and} \quad a = e^{\frac{-T}{RC}}$$

Now we have $H(z)$, i.e., yet another I/O relation for the discrete-time filter…

Based on the transfer function $H(z)$, we can now find the difference equation…

$$X(z) \longrightarrow \boxed{H(z)} \longrightarrow Y(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b}{1 - az^{-1}}$$

$$Y(z)(1 - az^{-1}) = bX(z)$$

$$Y(z) = aY(z)z^{-1} + bX(z)$$

$$y[n] = \mathcal{Z}^{-1}\{Y(z)\} = ay[n-1] + bx[n]$$

Voila… Now we have the time domain I/O relation for the discrete-time filter…

This is called the constant coefficient difference equation.

Useful…???

Yes…! The difference equation is extremely useful

$$y[n] = ay[n-1] + bx[n]$$

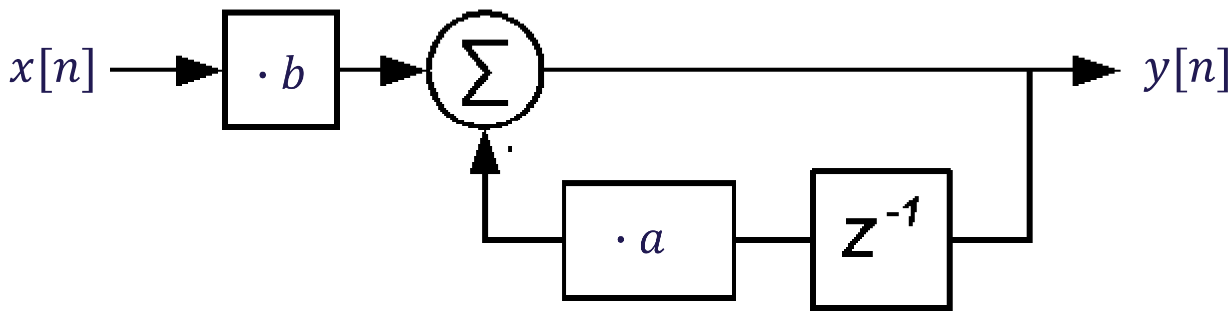
Using this expression, we can now easily create our much wanted program in terms of an infinite loop;

```
y_old := 0;                    /* Initialize the variable y_old
START                          /* Start label
  x := ADC;                    /* Read the next value from the ADC
  y_new := a*y_old + b*x;      /* Calculate the difference equation
  DAC := y_new;                /* Write the result to the DAC
  y_old := y_new;              /* Update the variable y_old
GOTO START                     /* Jump to start
```

Wow… This little program constantly calculates the convolution between the input sequence $x[n]$ and the impulse response $h[n]$.

We could also express the difference equation in graphical form;

$$y[n] = ay[n-1] + bx[n]$$



This is an example of a recursive digital filter, i.e., its impulse response will never become zero…

Therefore, such a filter is also known as an Infinite Impulse Response (IIR) filter.

So, we made it… We transformed our little RC-based continuous-time1'st order Butterworth lowpass filter into a discrete-time equivalent…!!

And the method we used is called the Impulse Invariant Method