Digital Signal Processing
ESD-5 and IV-5/Elektro, E24

1. Introduction to DSP and Digital Filters,
- The Impulse Invariant Method

Assoc. Prof. Peter Koch, AAU

Welcome to the course Digital Signal Processing

Viden

- Teori og metoder til non-parametrisk spektralestimering, herunder f.eks. Diskret Fourier Transformation (DFT) og dennes realisation i form af Fast Fourier Transformation (FFT) og Short Time Fourier Transformation (STFT).
- Sammenhænge mellem tids- og frekvensdomænerepræsentationer, herunder f.eks. Heisenberg's usikkerhedsrelation.
- Vinduesfunktioner og disses karakteristika i tids- og frekvensdomænet.
- Metoder til konstruktion af digitale filtre med såvel endelig som uendelig impulsrespons, herunder realisationsstrukturer og disses beregningsmæssige karakteristika.
- Group delay, minimum- og lineær fase systemer samt f.eks. all-pass og half-band filtre.
- Re-sampling og multirate filtre.
- Talrepræsentationer anvendt i realtids digital signalbehandling, herunder også f.eks. kvantisering, skalering, signal-støj forhold (SNR) og støj-optimale strukturer.
- CPU-strukturer til signalbehandlings-algoritmer, herunder f.eks. aritmetiske enheder samt kontrol- og hukommelses-organisation.

Færdigheder

- Anvende relevante værktøjer som f.eks. Matlab eller Python til spektralestimering.
- Anvende relevante værktøjer til design, simulering og realtids-implementering af digitale filtre, herunder også mere komplekse algoritmer som f.eks. multirate-strukturer.
- Vurdere betydningen af kvantisering samt anvende skalering ved design og implementering af filter-strukturer, herunder f.eks. beregning af SNR på filterets udgang.
- Vurdere hvorvidt forskellige signalbehandlings-algoritmer opfylder givne specifikationer i såvel tids- som i frekvensdomænet.

Kompetencer

- Den studerende har kompetencer til at analysere signaler samt implementere digitale signalbehandlingsløsninger til f.eks. bortfiltrering af støj eller equalizing af signaler med brug af relevante værktøjer under hensyntagen til praktiske aspekter i implementeringen såsom endelig ordlængde.

Preliminary course overview

- We have 12 lectures scheduled, one per week, from Sep. 6 to Nov. 22.
- We will read selected parts – i.e., not all – of our textbook to cover the topics mentioned in the Study Regulation.
- Each lecture consists of an oral presentation of “Today’s Topic”, followed by an exercise session.
- The course is 5 ECTS, i.e., 150 hours.
  - ❖ Lectures & Exercises, 2 ECTS
  - ❖ Self study, 1 ECTS
  - ❖ Mini-project, 1 ECTS
  - ❖ Preparation for the exam, 1 ECTS
- The mini-project is a limited problem which must be solved by the groups. The findings are presented in a short technical report, 12-15 pages.
- The exam is oral with internal censor – 25 min. per student. Before the end of the course, you will be informed about the topics you can draw. You will be given 25 min. for preparation prior to the exam. At the exam, we will discuss the topic you have drawn, and the findings you have presented in your mini-project.

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., way more than 100 years – what is done with analog circuits can be considered as signal processing, typically filtering, modulation…

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 also address these methods in this course).

AALBORG UNIVERSITY
DENMARK

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, published the paper ”The Columbia Long-Playing Microgroove Recording System”. The LP record lasted for two generarions – and has now gained renewed interests among enthusiasts…

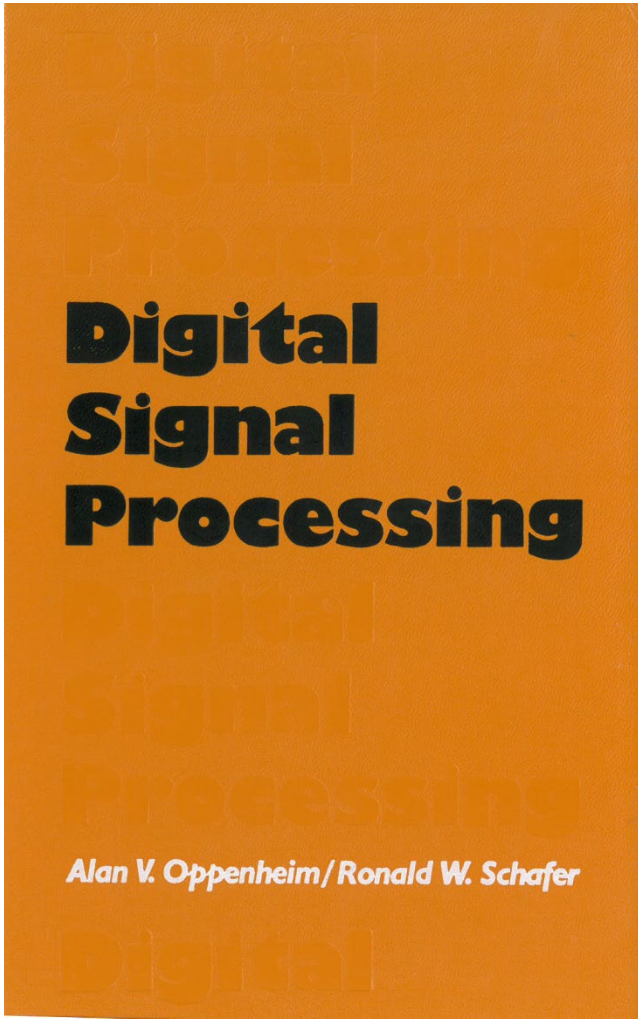
In the signal processing community, 1948 is therefore know as the ”Annus Mirabilis”…

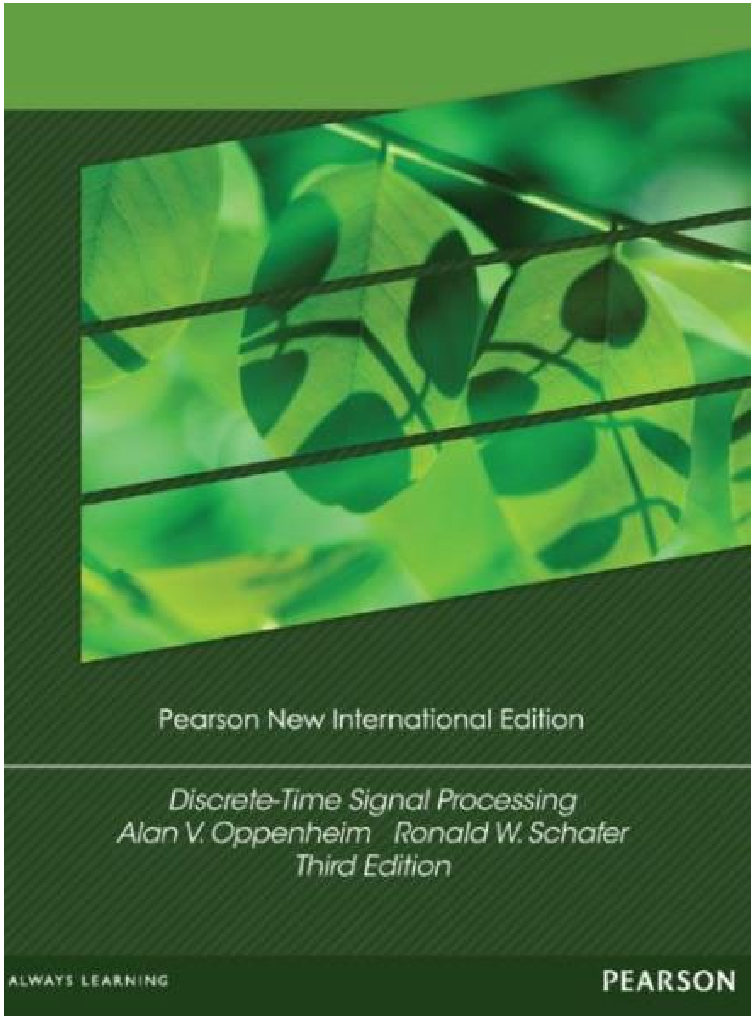
During the 50’s, 60’s and 70’s many mathematical theories and methods which are fundamental to modern discrete-time (digital) signal processing are being developed.

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

Oppenheim & Schafer published 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)





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…

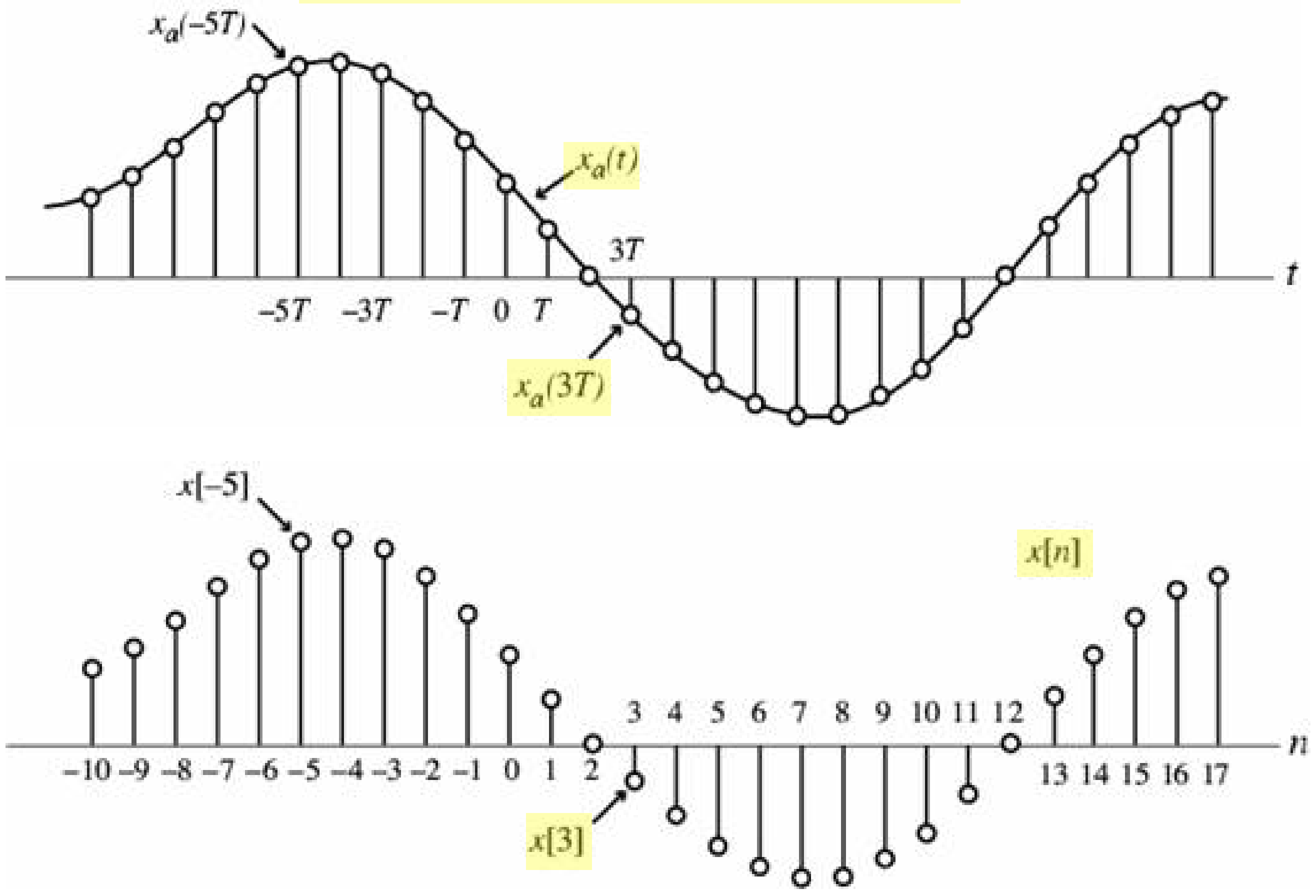
The organization Institute of Electrical and Electronics Engineers (IEEE) has produced a short video which tries to envision the way signal processing has penetrated our daily life…

AALBORG UNIVERSITY
DENMARK

…and now a few essential recaps from some of your previous courses related to Digital Signal Processing…

AALBORG UNIVERSITY
DENMARK

A discrete-time signal is derived by sampling a continuous-time signal…



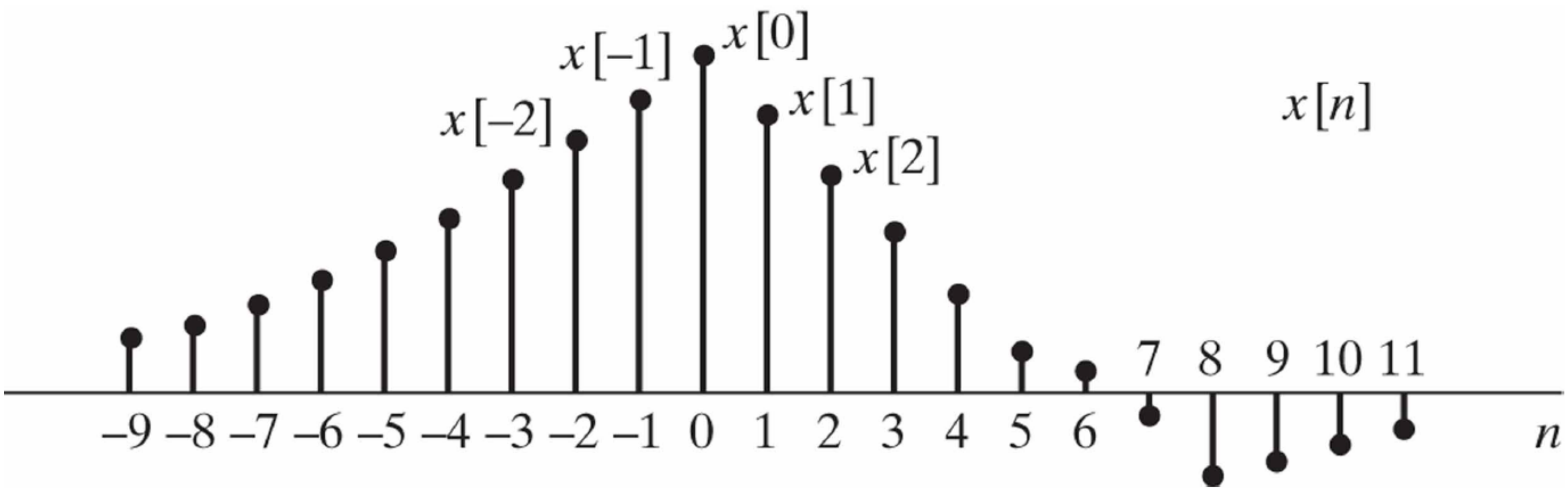
The continuous-time signal is sampled with a frequency known as the ”sample frequency”, f_s. The reciprocal of the sample frequency is known as the ”sample period”, $T = {}^{1}/_{f_s}$. This leads to equidistant sampling.

AALBORG UNIVERSITY
DENMARK

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

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

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

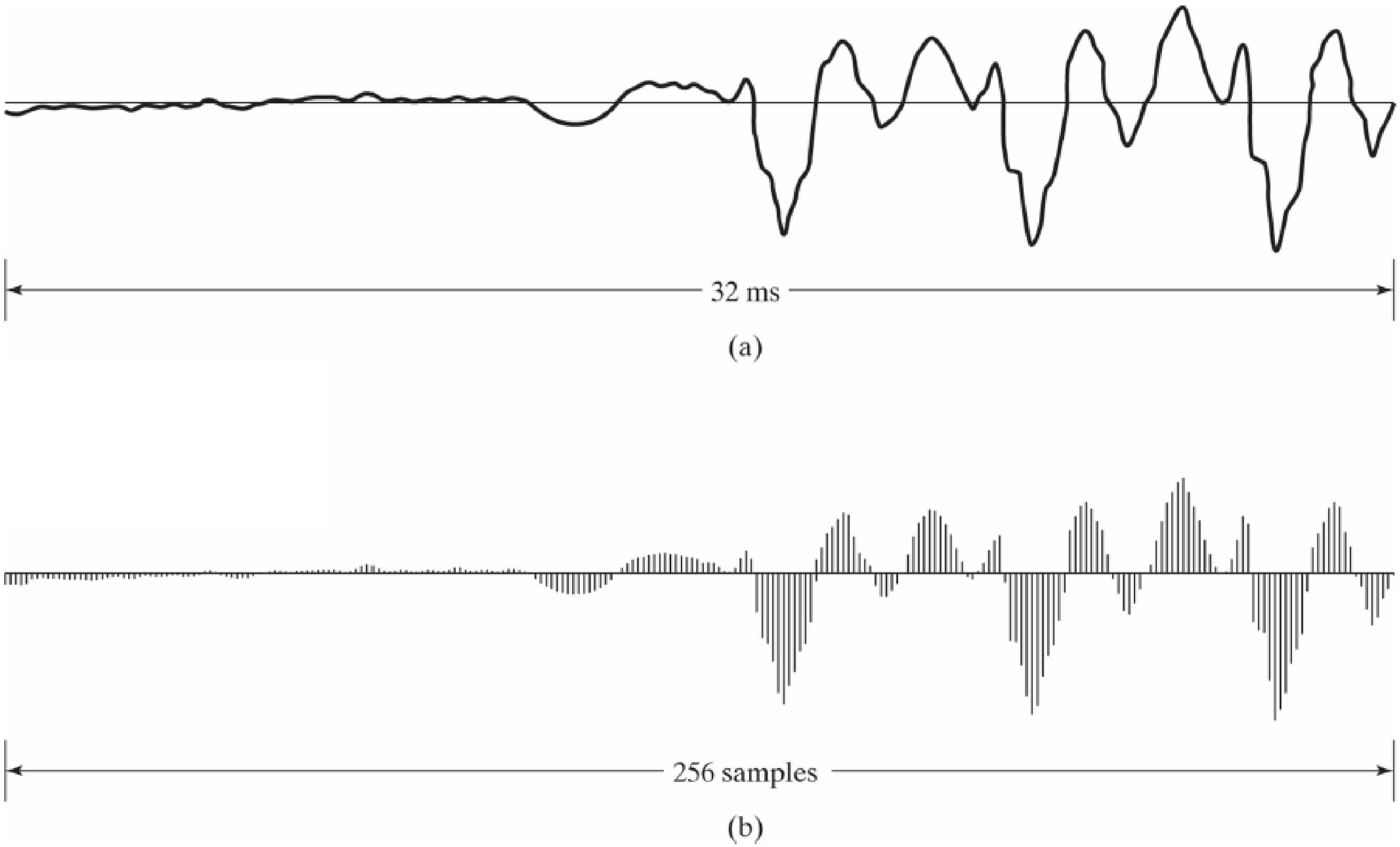


$x[n]$ is defined only for the values n.

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

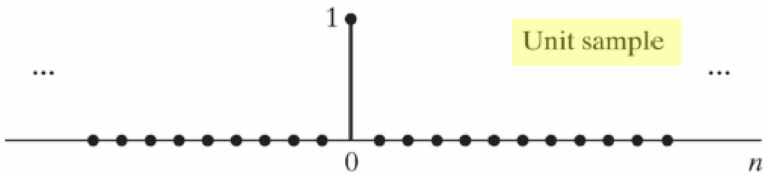
AALBORG UNIVERSITY
DENMARK

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



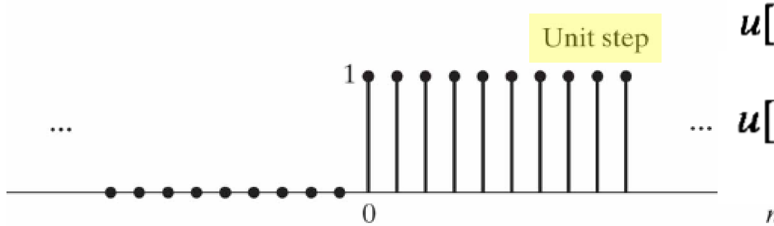
Important sequences…

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



(a)

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

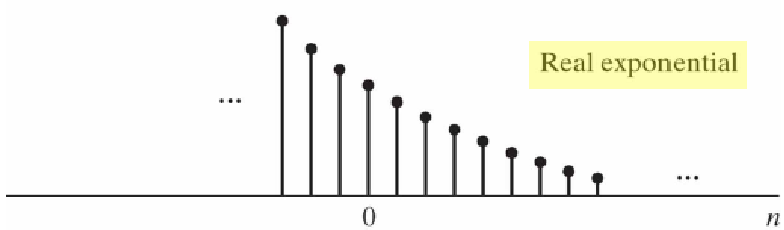


(b)

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

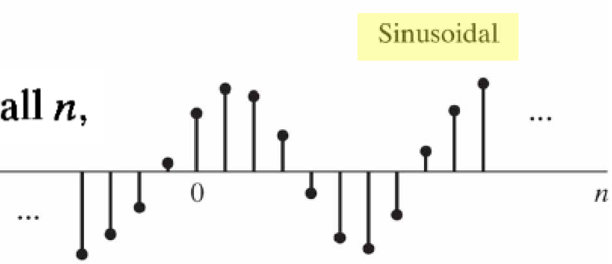
$$u[n] = \sum_{k=0}^{\infty} \delta[n-k].$$

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



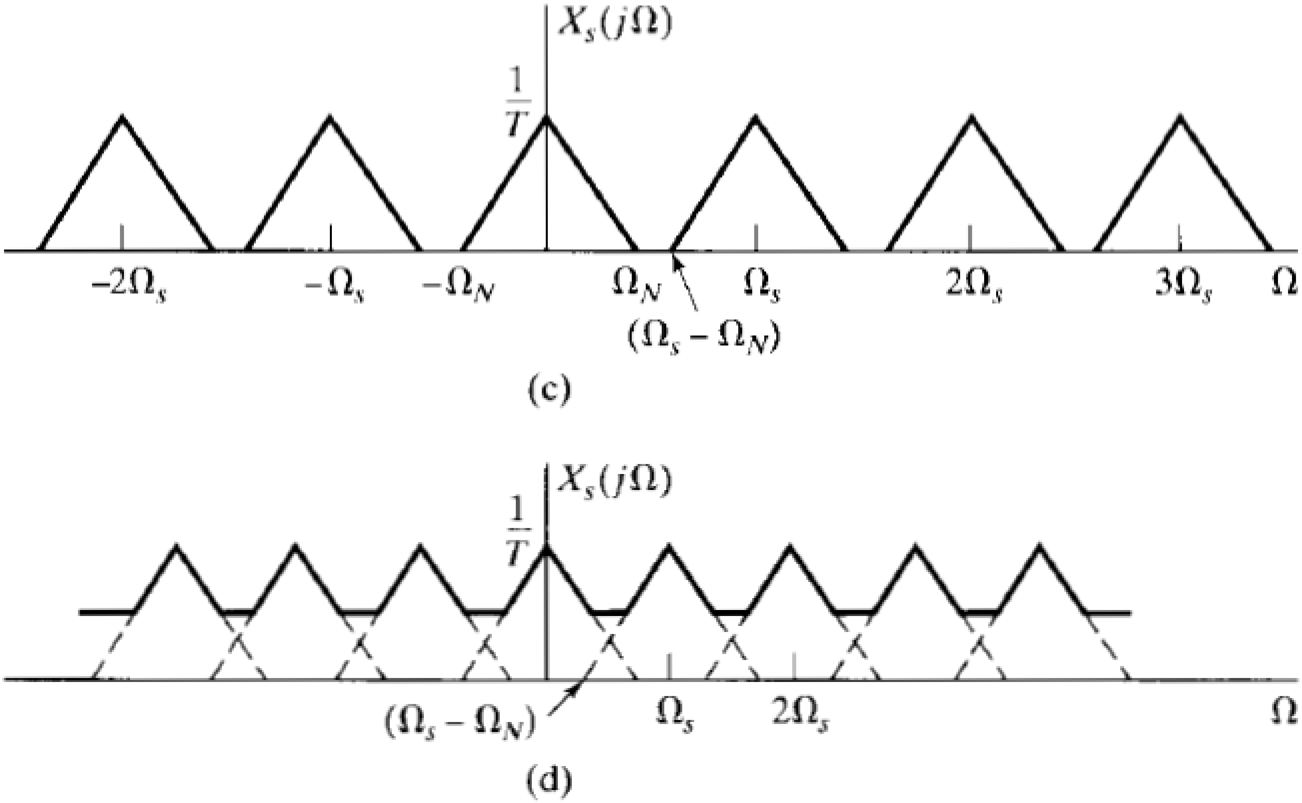
(c)

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



(d)

When the signal becomes discrete in the time-domain, it becomes periodic in the frequency-domain – and this potentially leads to a serious problem; Aliasing…!



Here $X_c(j\Omega)$ can be reconstructed by using an ideal LP filter with $\Omega_{cut} = {}^{\Omega_s}/_{2}$.

Here $X_c(j\Omega)$ cannot be reconstructed by LP filtering.

AALBORG UNIVERSITY
DENMARK

Therefore, always comply with Nyquist Sampling Theorem

Let $x_c(t)$ be a bandlimited signal with

$$X_c(j\Omega) = 0 \qquad \text{for } |\Omega| \geq \Omega_N.$$

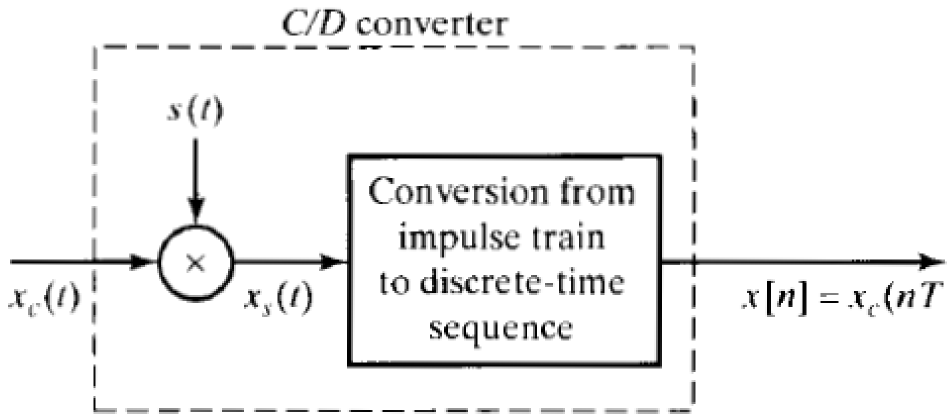
Then $x_c(t)$ is uniquely determined by its samples $x[n] = x_c(nT)$, $n = 0, \pm 1, \pm 2, \ldots,$ if

$$\Omega_s = \frac{2\pi}{T} \geq 2\Omega_N.$$

The frequency Ω_N is commonly referred to as the *Nyquist frequency*, and the frequency $2\Omega_N$ that must be exceeded by the sampling frequency is called the *Nyquist rate*.

AALBORG UNIVERSITY
DENMARK

Seen from a mathematical point of view, sampling consists of two separate functions – modulation and generation of a sequence.



Based on this model we can derived the periodicity property of $\mathcal{F}\{x_s(t)\}$

$X_s(j\Omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s))$, equ. 6 on p. 166.

…and similarly we can find an expression for $\mathcal{F}\{n[n]\}$

$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - k\frac{2\pi}{T}\right)\right)$, equ. 20 on p. 171.

AALBORG UNIVERSITY
DENMARK

$X_s(j\Omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s))$, equ. 6 on p. 166.

$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - k\frac{2\pi}{T}\right)\right)$, equ. 20 on p. 171.

Note that we use Ω and ω, respectively, for the CT and the DT frequency…!!

Comparing these two expressions, we see that $X(e^{j\omega})$ is simply a frequency-scale version of $X_s(j\Omega)$ with the frequency scaling specified by $\omega = \Omega T$.

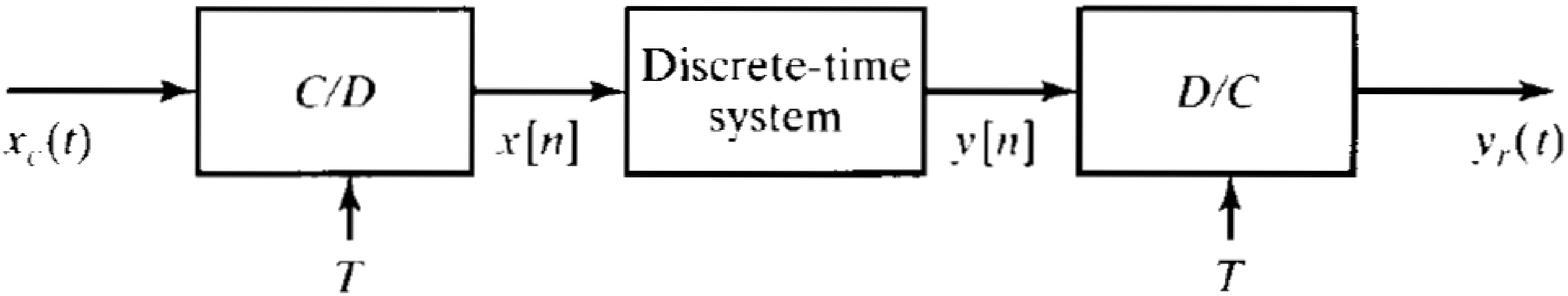
$$\boxed{\omega = \Omega T}$$

We also recognize that $\omega = 2\pi$ corresponds to the Sample Frequency…!!

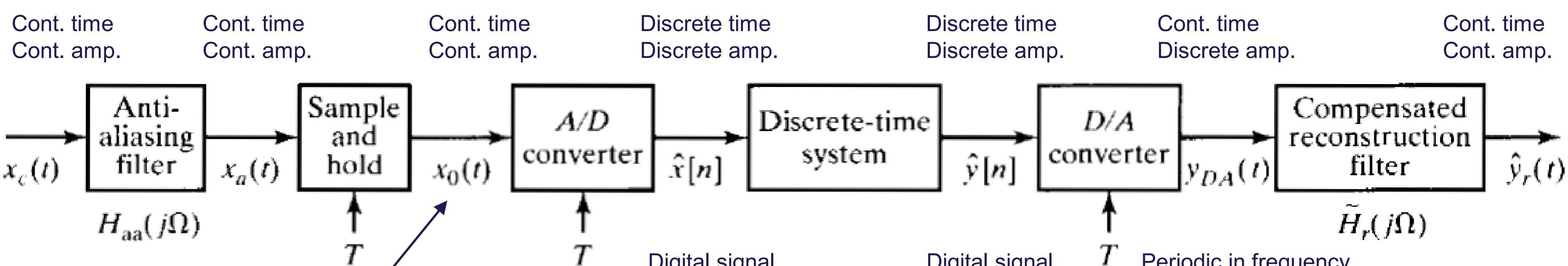
So, in conclusion, EVERYTHING in the discrete-time domain is 2π periodic and is normalized to the Sample Frequency…!!

DTS Systems – the overall structure

In the ideal situation, a discrete-time signal processing system should consist of three fundamental functions (or blocks).



The problem is however, that it is not possible to construct an ideal C/D (as we have just discussed), nor an ideal D/C. We must rely on an approximate realization…



"Sampling" means that the signal becomes discrete in time, and thus being known only at $t = nT$. However, since the signal is being held at the output of the S/H unit, then from a strict mathematical sence, the signal is still time-continuous – thus the notation $x_0(t)$.

Discrete-Time Filters

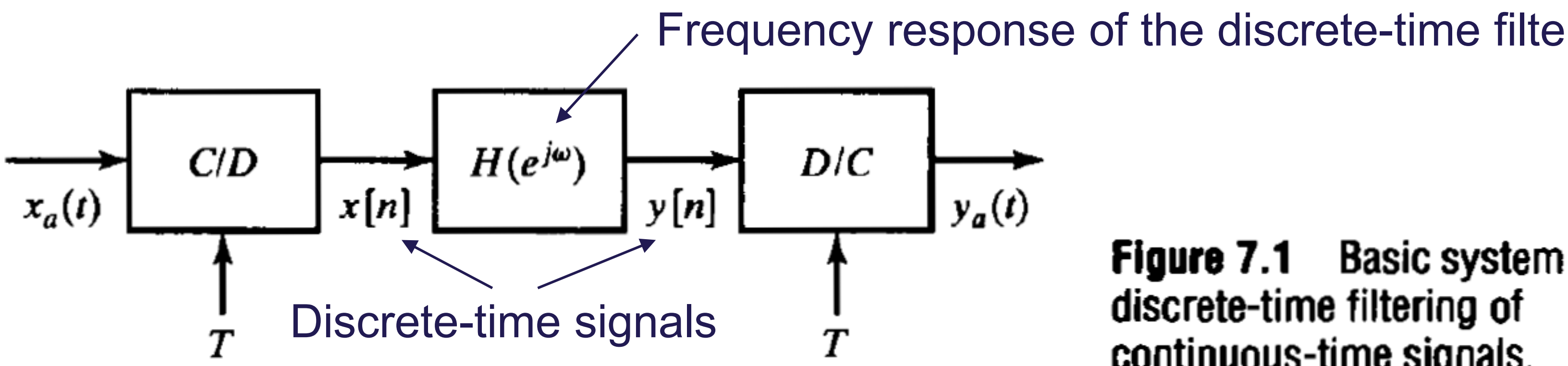


Figure 7.1 Basic system for discrete-time filtering of continuous-time signals.

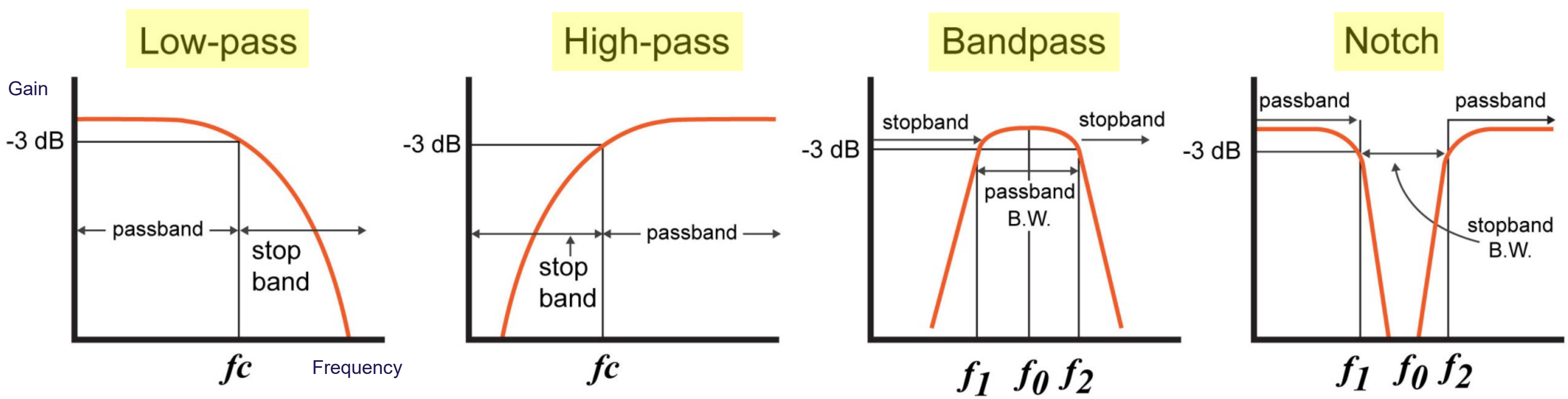
Specification of the Effective Filter; $H_{\text{eff}}(j\Omega) = \begin{cases} H(e^{j\Omega T}), & |\Omega| < \pi/T, \\ 0, & |\Omega| > \pi/T. \end{cases}$

This expression says that $H_{eff}(j\Omega)$ should be bandlimited to half the sample frequency

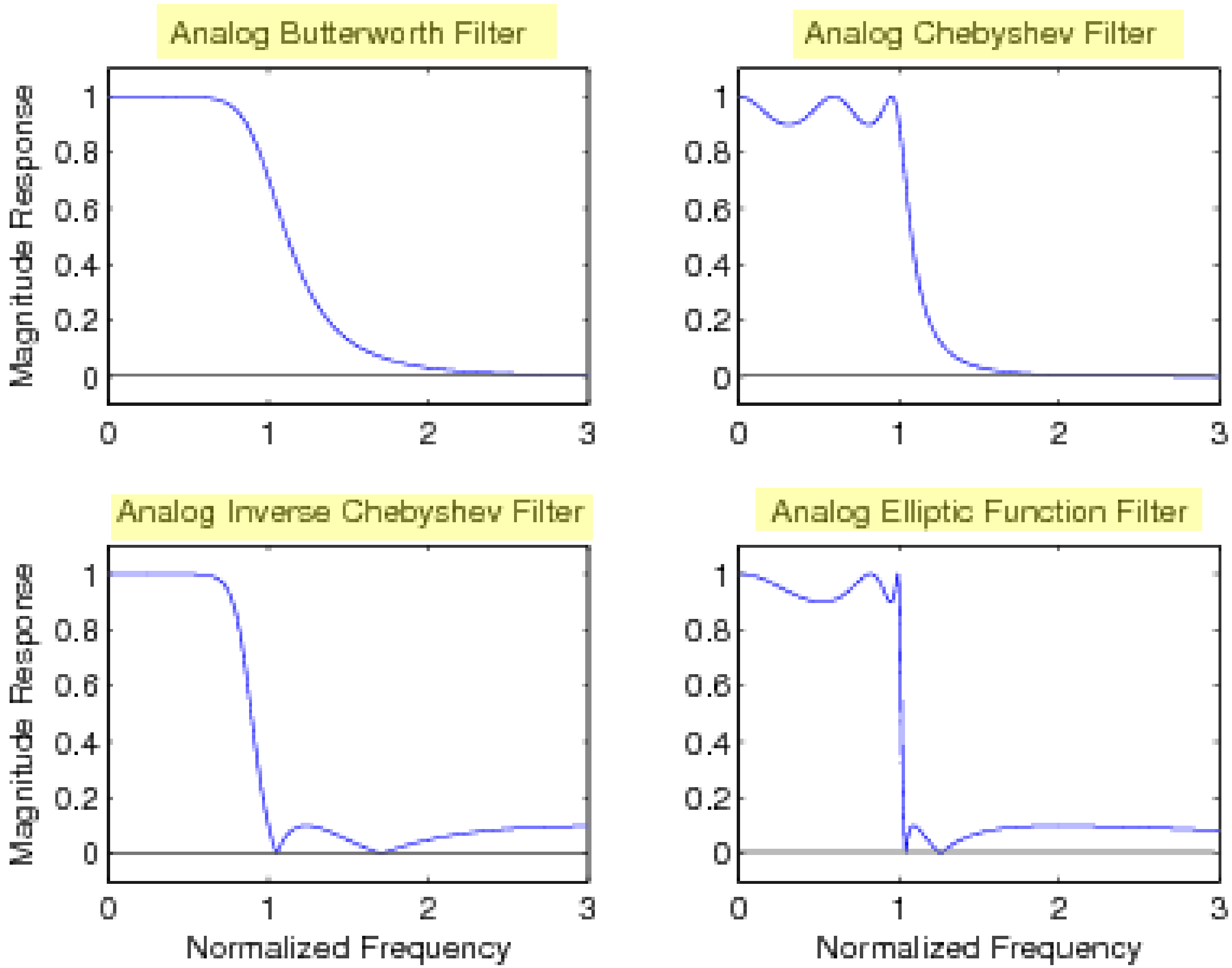
Specification of the discrete-time filter; $H(e^{j\omega}) = H_{\text{eff}}\left(j\frac{\omega}{T}\right), \qquad |\omega| < \pi.$

AALBORG UNIVERSITY
DENMARK

But what is the ”Effective Filter”…?



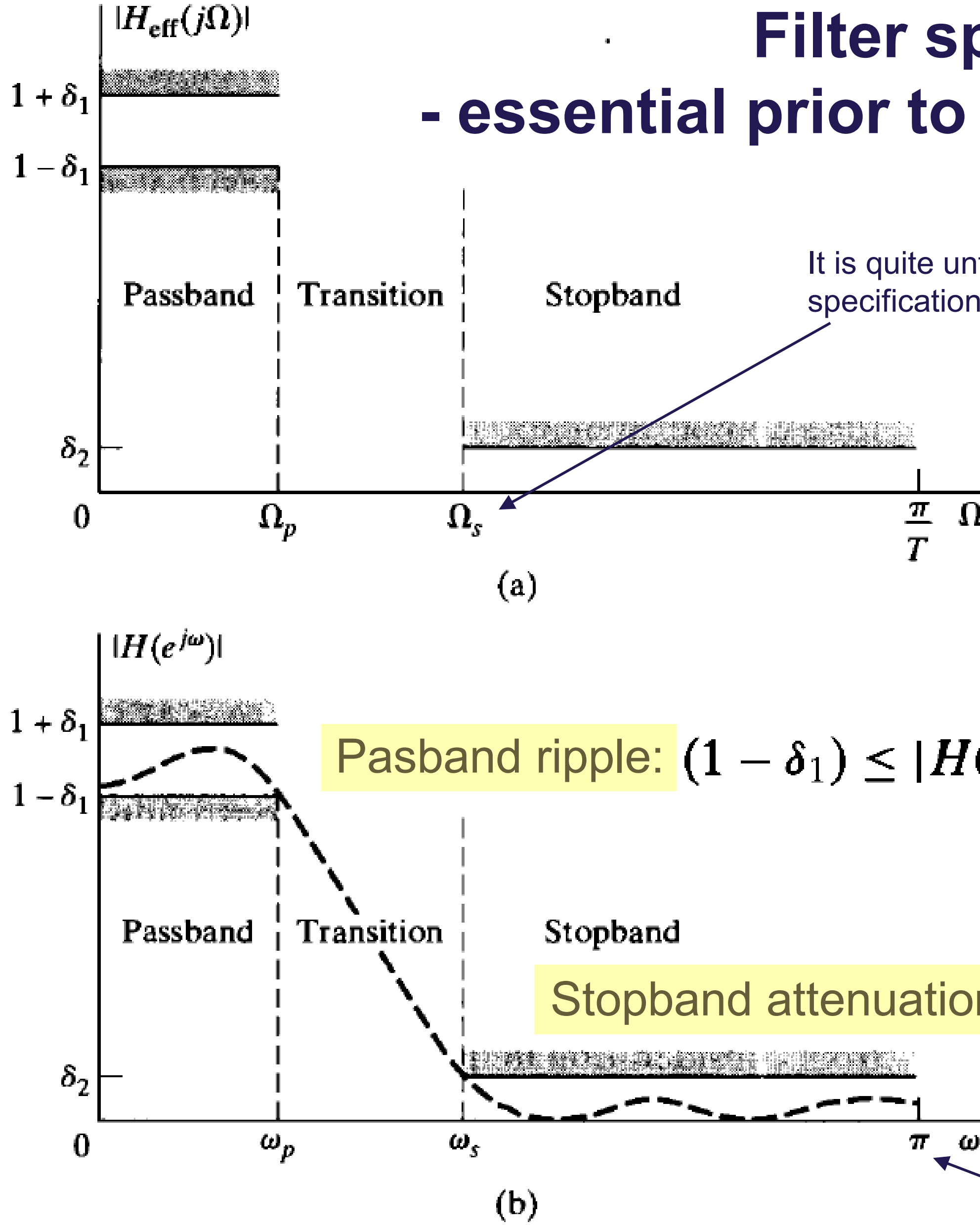
Frequency selective filters – here lowpas in various matematical configurations



AALBORG UNIVERSITY
DENMARK

Filter specification
- essential prior to the design of the filter

It is quite unfortunate that he uses Ω_s for the Stopband specification – he also uses Ω_s for the sample angle frequency..!



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

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

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.

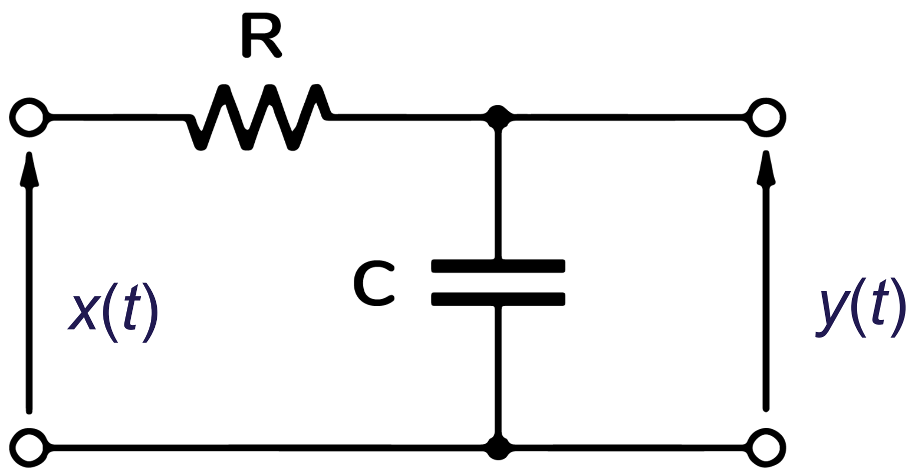
In the discrete-time domain, everything is related to the sample freqency 2π and thus also π.

Break…!

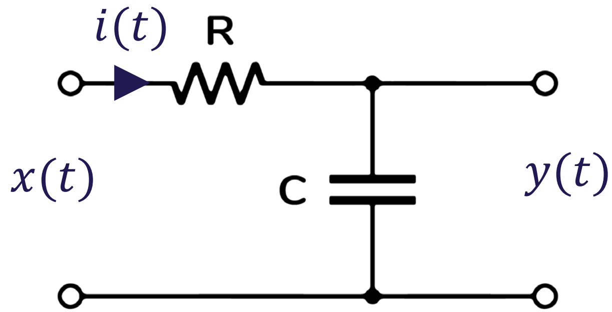
AALBORG UNIVERSITY
DENMARK

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

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

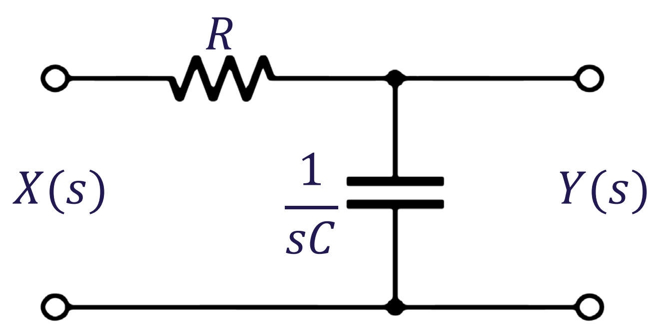
AALBORG UNIVERSITY
DENMARK

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

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;

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

AALBORG UNIVERSITY
DENMARK

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

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

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

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

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

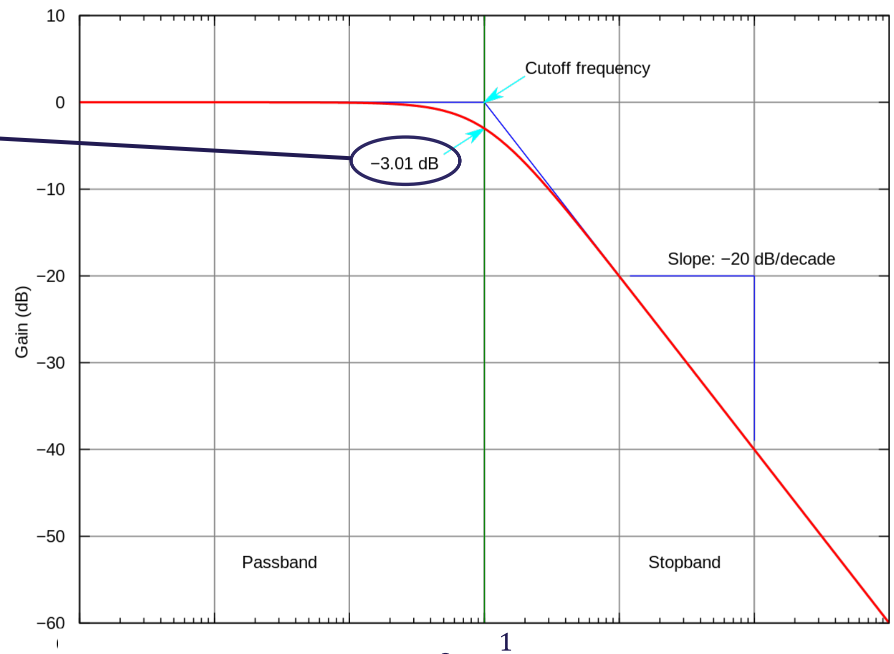
From the transfer function as well as from the differential equation we see that the parameter *RC* appears, and thus we may conclude that it impacts the overall working of the circuit. Let's look at the Amplitude response, i.e., $s = j\Omega$ in $H(s)$;

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

Frequency response

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

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

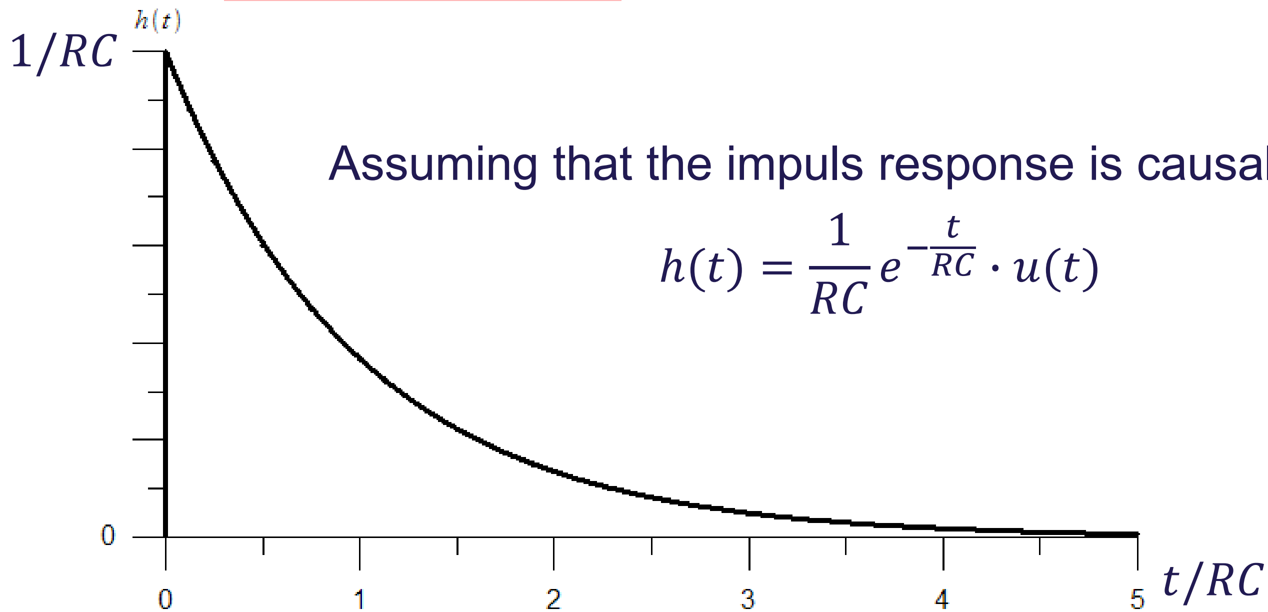


$\Omega = \frac{1}{RC}$

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

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

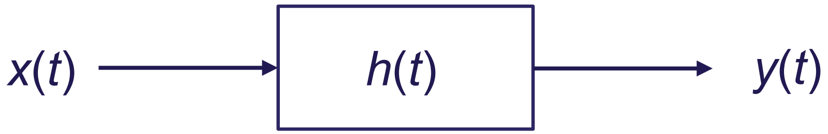
which is known as the Impluse response.



Assuming that the impuls response is causal, we get;

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

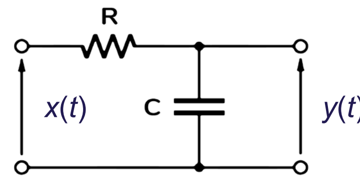
Now, given the impulse response, we can write the Convolution integrale;



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

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

Wow…! Think about it – this little circuit



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

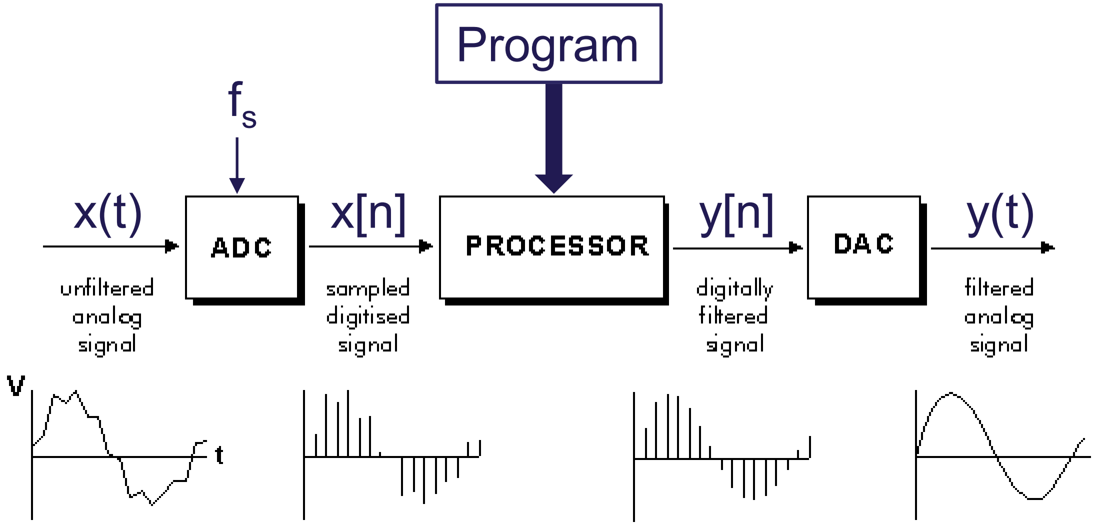
So, in conclusion we can state;

Given $H(s)$ or equivalently $h(t)$, then we have the totale I/O relation for the Linear Time-Invariant (LTI) system – a 1'st order RC LP-filter.

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 the output behaves just as the RC circuit..?

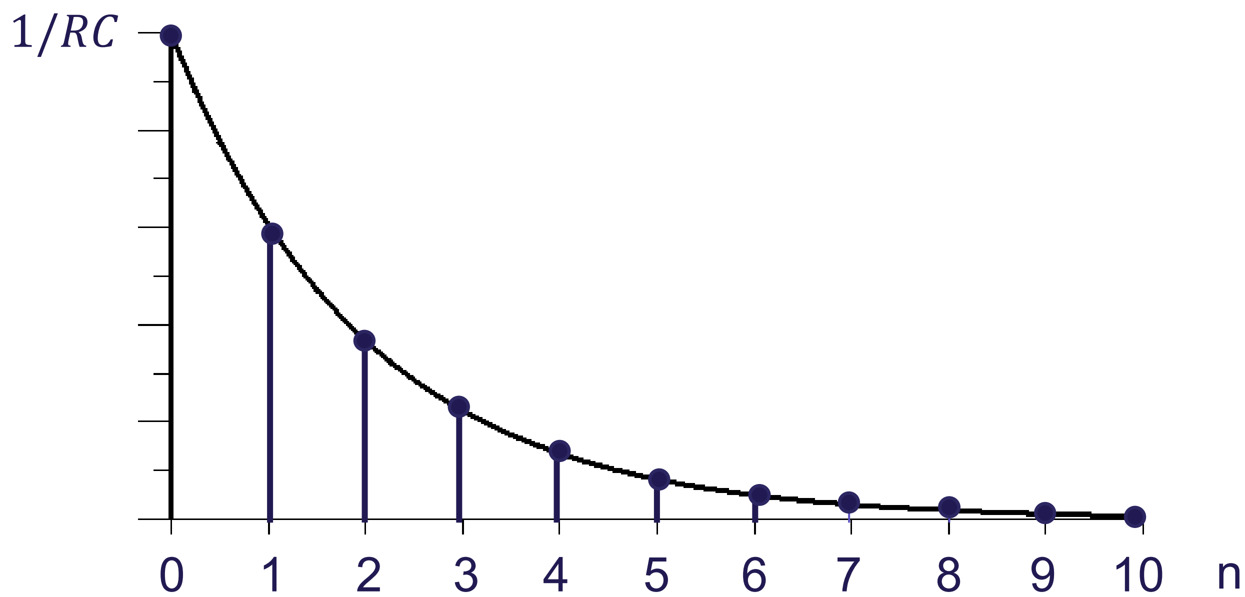
What is a signal processing system...



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.

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

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

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

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

$$= \frac{T}{RC} \sum_{k=0}^{\infty} e^{\frac{-kT}{RC}} \cdot x[n-k]$$

Useful…???

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

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

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

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

The system is Causal

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

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

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

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

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

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

…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;

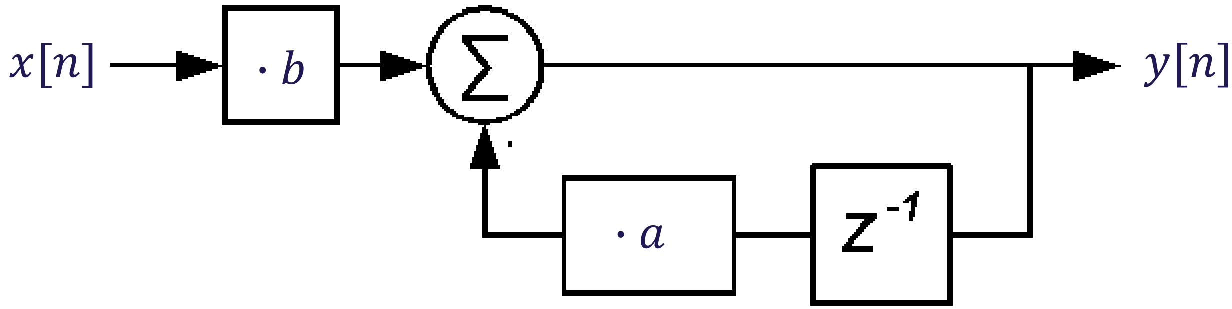
```
y_old := 0;                  /* Initialize the variable y_old
START                        /* Start label
  x := ADC;                  /* Read the next value from the ADC
  y_new := a*y_old + b*x;    /* Calculate the difference equation
  DAC := y_new;              /* Write the result to the DAC
  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 $h[n]$.

AALBORG UNIVERSITY
DENMARK

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