Université de Liège

ELEN0019:

Audio signal processing. Principles and experiments

Course objectives

- To give an introduction to digital signal processing in audio applications,

- Emphasis is on real-time applications, through the programming of a DSP (digital signal processor),

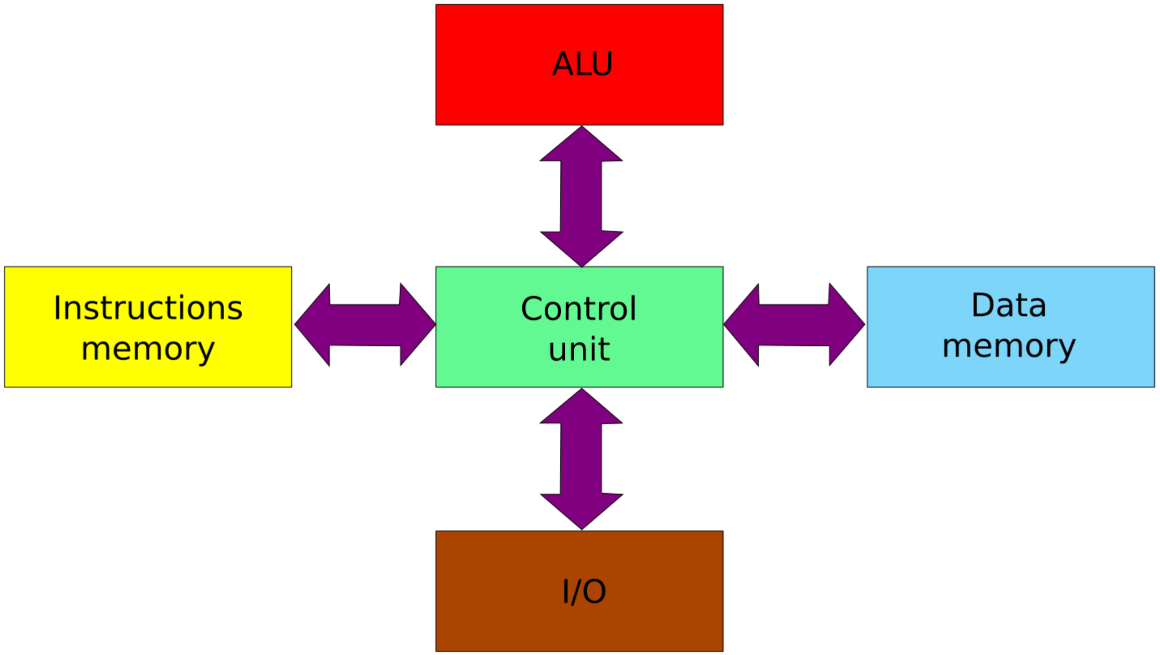
- Laboratory sessions: the students will have the opportunity to design their own systems, to test their operation through measurements and listening !

- Textbook: http://www.montefiore.ulg.ac.be/~josmalskyj/dsp.php

Digital signal processors (I)

- Specific processors optimized for real-time signal processing,
- Example: the MAC operation (multiply and accumulate), mostly used in the *convolution* operations,
- The *Harvard architecture* of the processor allows to read an instruction and the corresponding data in memory, in one clock cycle.

Digital signal processors (II)



The *Harvard architecture* is characterized by two blocks of memories: one for the program and one (two) for the data.

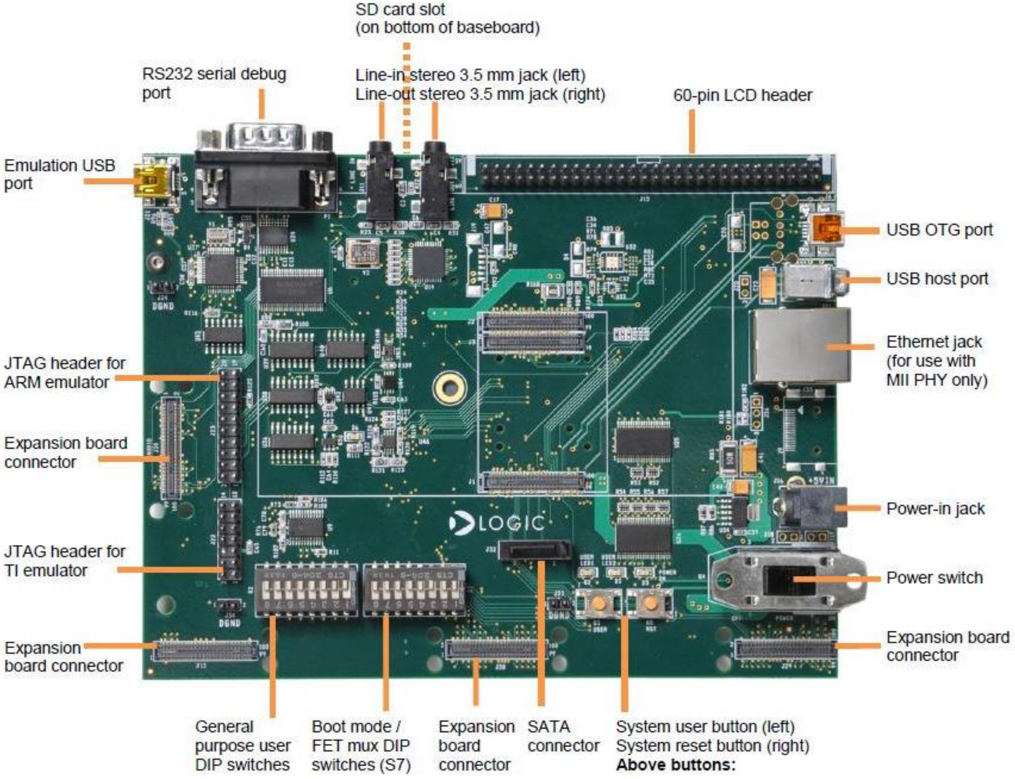
With separate buses.

Digital signal processors (III)

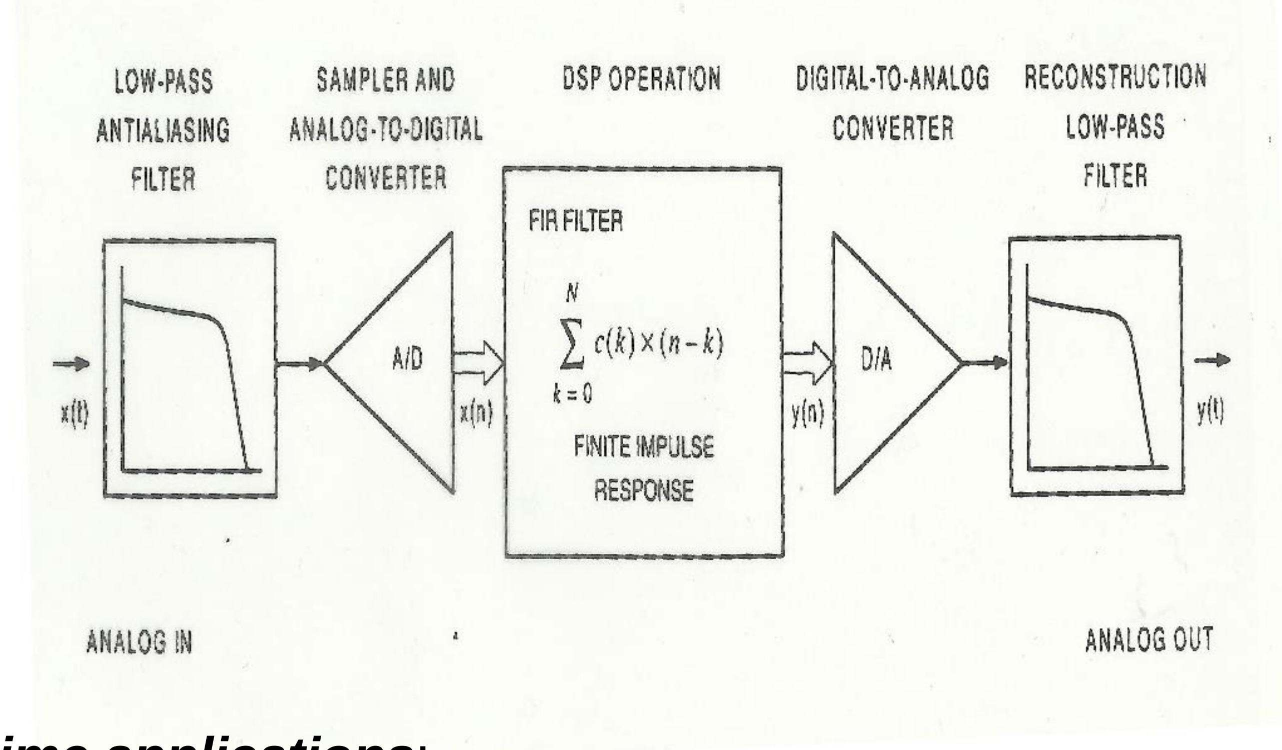
Texas C6748 DSP

- Floating-point/fixed point architectures,
- 32-bit registers (data/addreses),

- 1 stereo input/1 stereo output for audio,
- f_e = 8 to 96 kHz (codec),
- 16-bit audio samples.



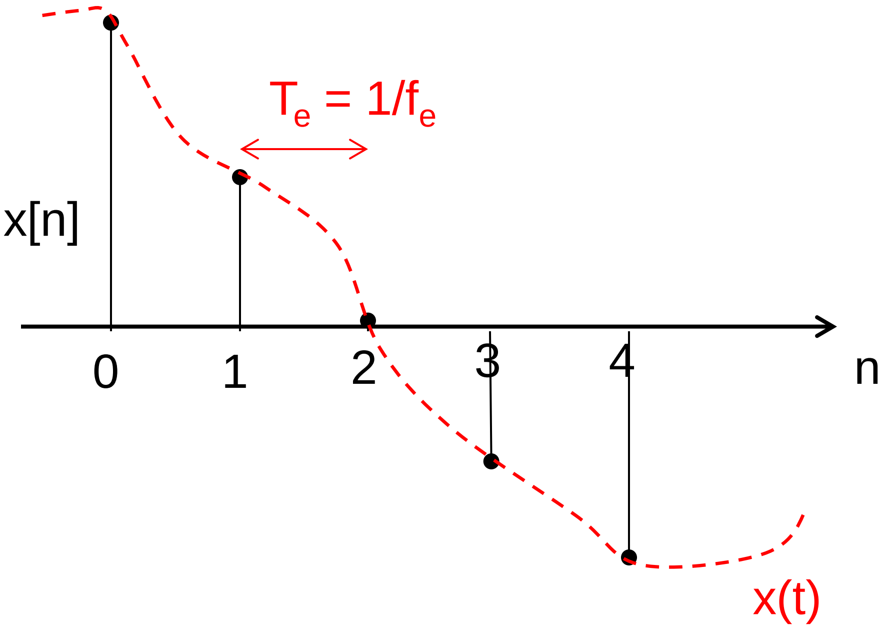
Digital signal processing



Real-time applications:

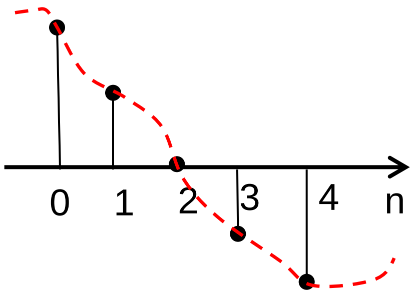
- each input sample must be processed (or stored) in one sampling period,
- a sample must be fed to the output before the end of each sampling period.

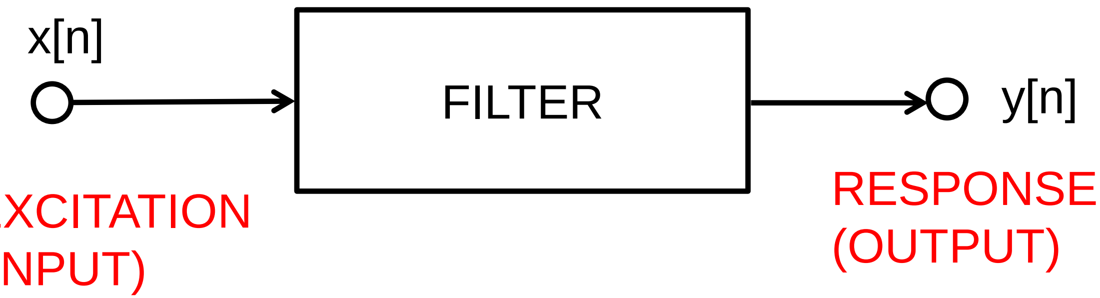
Digital (audio) signals



Sampling (Shannon) theorem: $f_e > 2.f_H$ to avoid aliasing.

Digital (audio) filters (I)





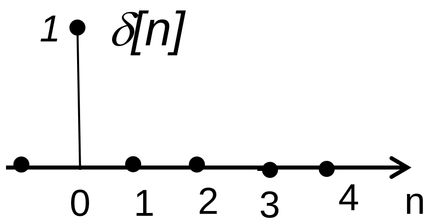
CAUSALITY: if $x[n]=0$ for $n<n_0$, then $y[n]=0$ for $n<n_0$

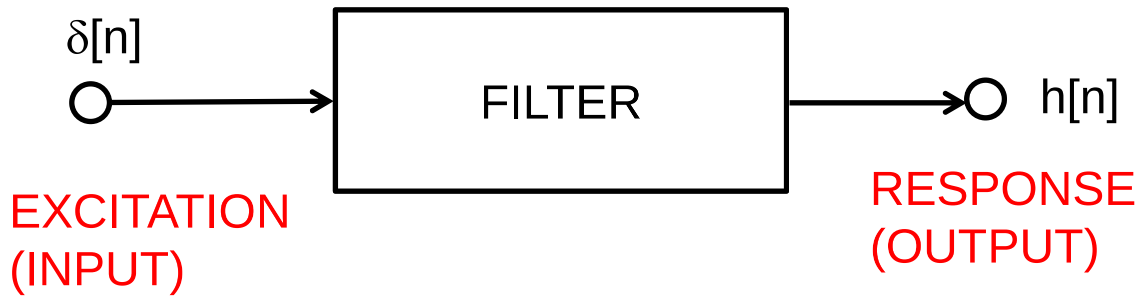
STABILITY : $y[n]$ bounded if $x[n]$ bounded

LINEAR AND TIME-INVARIANT (LTI) SYSTEMS:

$$b_0\, y[n] + \ldots\, b_N\, y[n-N] = a_0\, x[n] + \ldots\, a_M\, x[n-M]$$

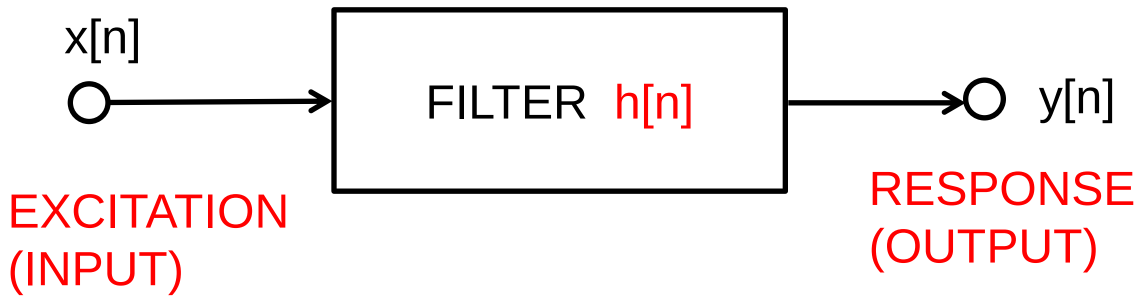
Digital (audio) filters (II)





IMPULSE RESPONSE: if $x[n]=\delta[n]$

Digital (audio) filters (III)



IMPULSE RESPONSE: h[n]

FOR LTI SYSTEMS:

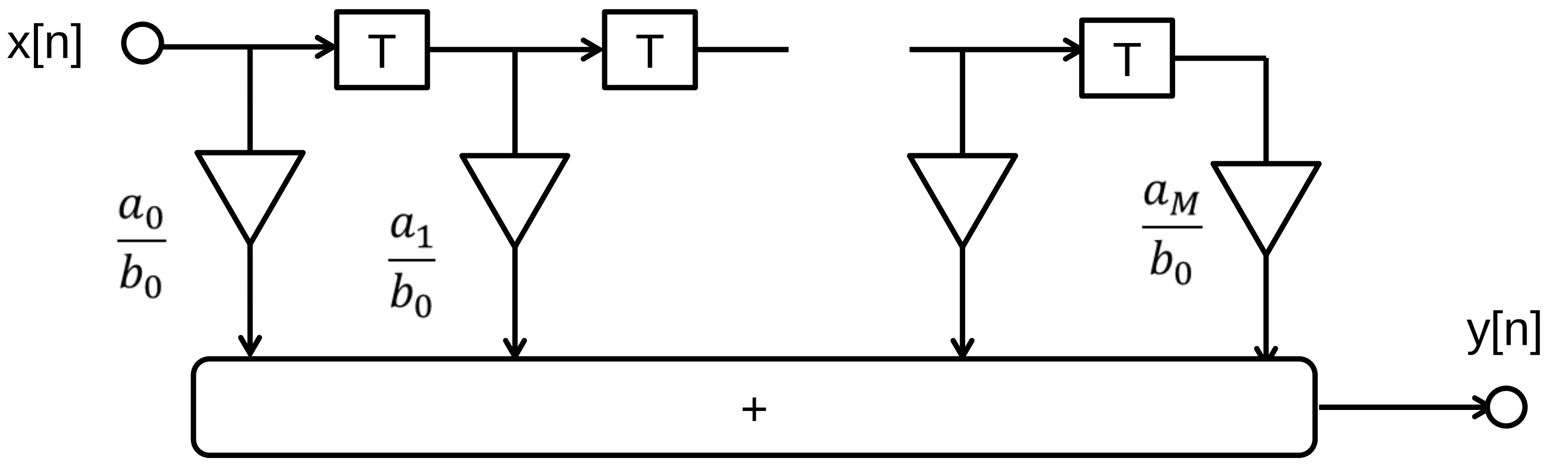
$$y[n] = x[n] * h[n] = \sum_{l=-\infty}^{+\infty} x[l].h[n-l]$$

STABILITY OF LTI SYSTEMS:

$$\sum_{l=-\infty}^{+\infty} |\, h[l] \,| < \infty$$

Digital (audio) filters (IV)

NON-RECURSIVE FILTER:

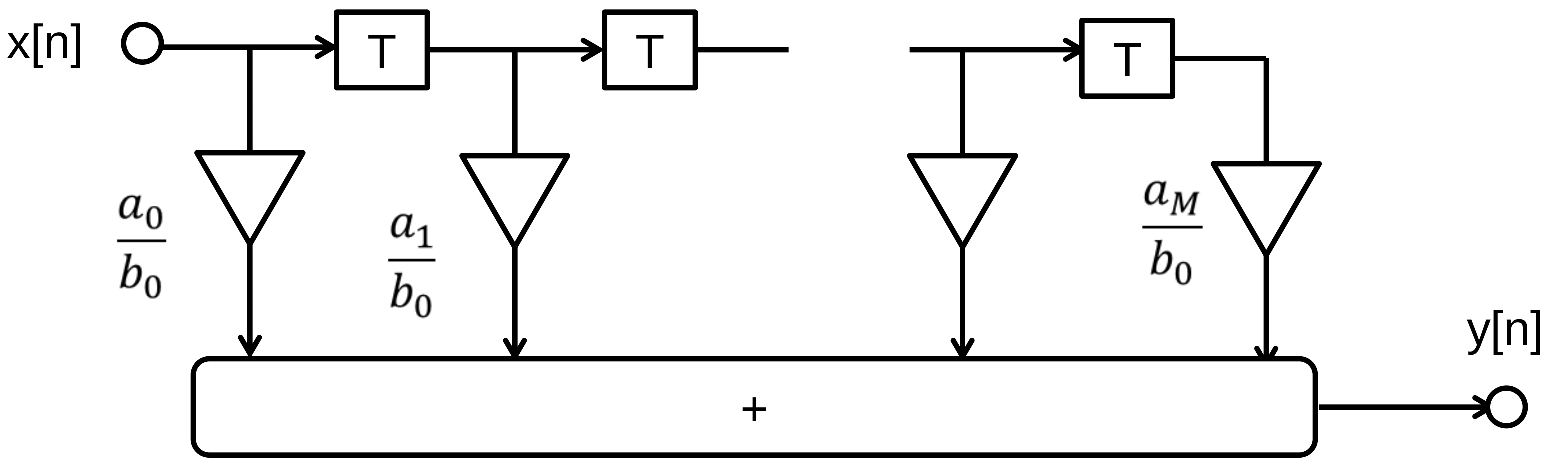


Input/output equation: $b_0\, y[n] = a_0\, x[n] + a_1\, x[n-1] + \dots\, a_M\, x[n-M]$

A non-recursive filter always has a finite impulse response: FIR: Finite Impulse Response

Digital (audio) filters (IV)

NON-RECURSIVE FILTER:



b_0 y[n] = a_0 x[n] + a_1 x[n-1] + … a_M x[n-M]

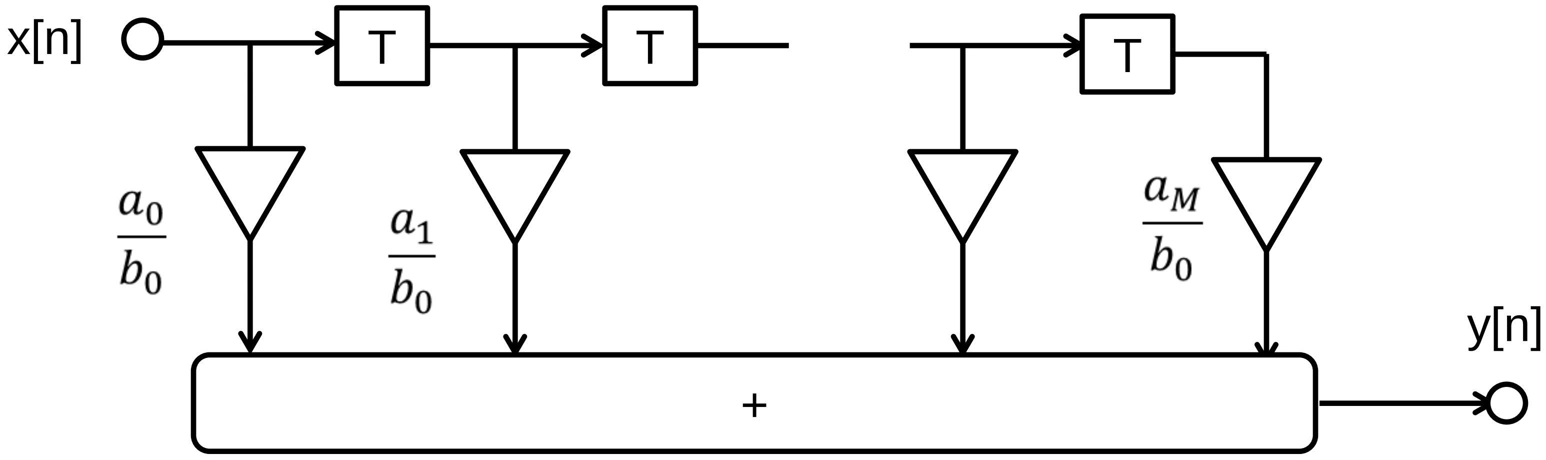
$$y[n] = x[n] * h[n] = \sum_{l=-\infty}^{+\infty} h[l].x[n-l]$$

coefficients

$a_i/b_0 = h[i]$

Digital (audio) filters (IV)

NON-RECURSIVE FILTER:

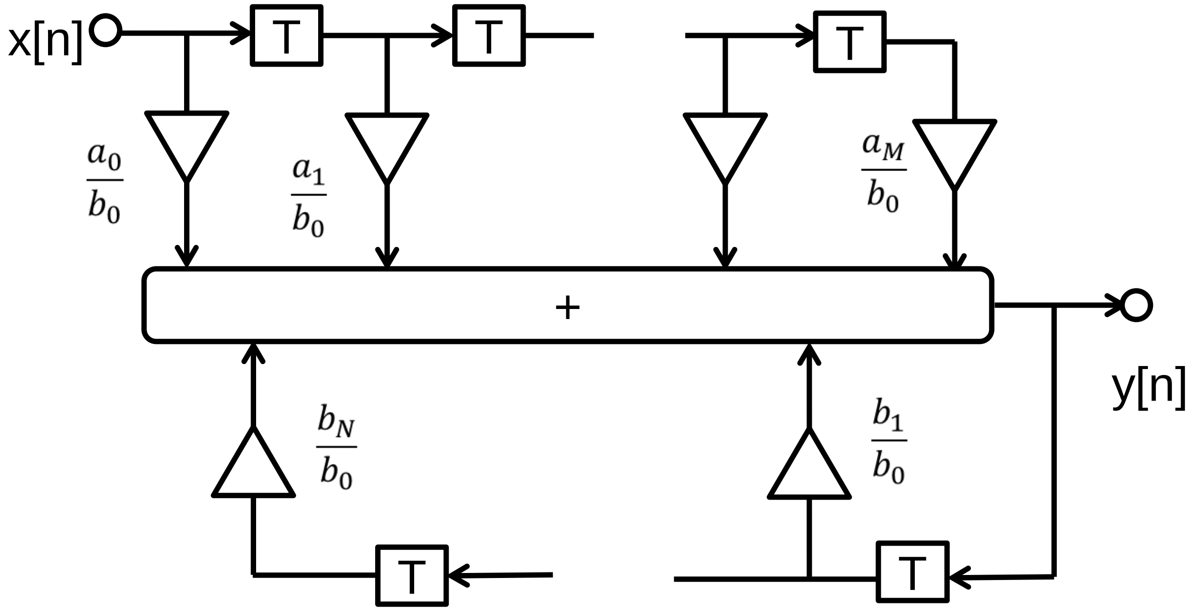


Input/output equation: $b_0\, y[n] = a_0\, x[n] + a_1\, x[n-1] + \dots\, a_M\, x[n-M]$

A non-recursive filter is always stable

Digital (audio) filters (V)

RECURSIVE FILTER:



b_0 y[n] = a_0 x[n] + a_1 x[n-1] + … a_M x[n-M] + b_1 y[n-1] + … b_N y[n-N]

If a filter has a <u>infinite</u> impulse response, then it is a recursive filter:
IIR: Infinite Impulse Response

z- Transform (I)

$$X(z) = \sum_{n=-\infty}^{\infty} x[n].z^{-n}$$

$$x[n-n_0] \rightarrow z^{-n_0} X(z)$$

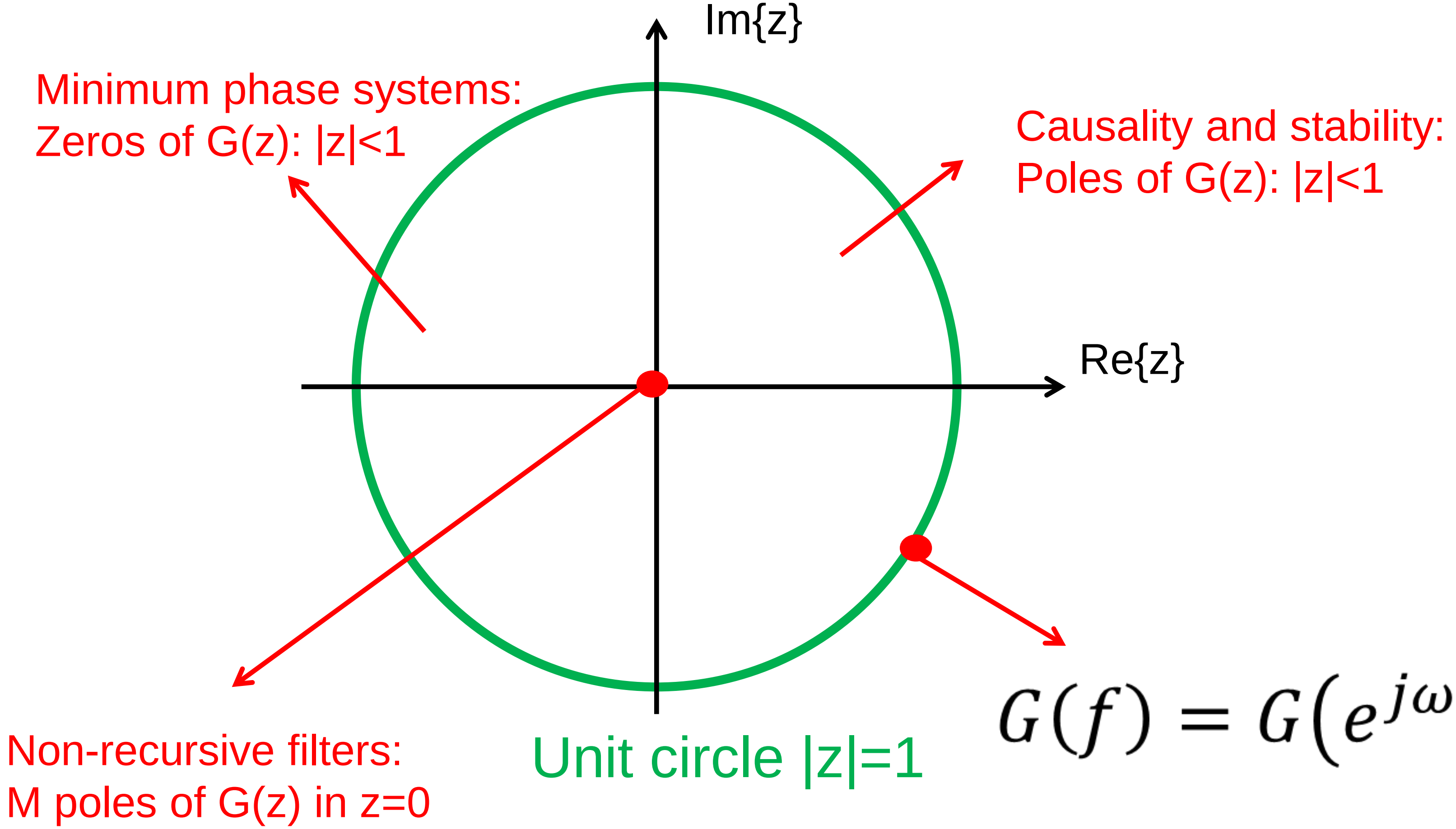
$$x[n] * y[n] \rightarrow X(z).Y(z)$$

TRANSFER FUNCTION OF A LTI SYSTEM:

b_0 y[n] + b_1 y[n-1] + … b_N y[n-N] = a_0 x[n] + a_1 x[n-1] + … a_M x[n-M]

$$G(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1.z^{-1} \ldots + a_M.z^{-M}}{b_0 + b_1.z^{-1} \ldots + b_N.z^{-N}}$$

z- Transform (II)

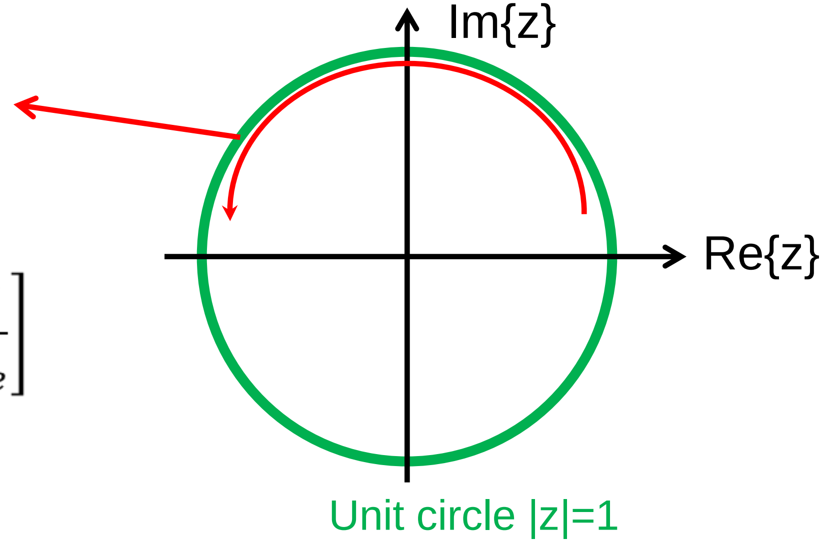


$$G(f) = G\left(e^{j\omega T_e}\right)$$

Frequency response

How to obtain the frequency response from the transfer function G(z) ?

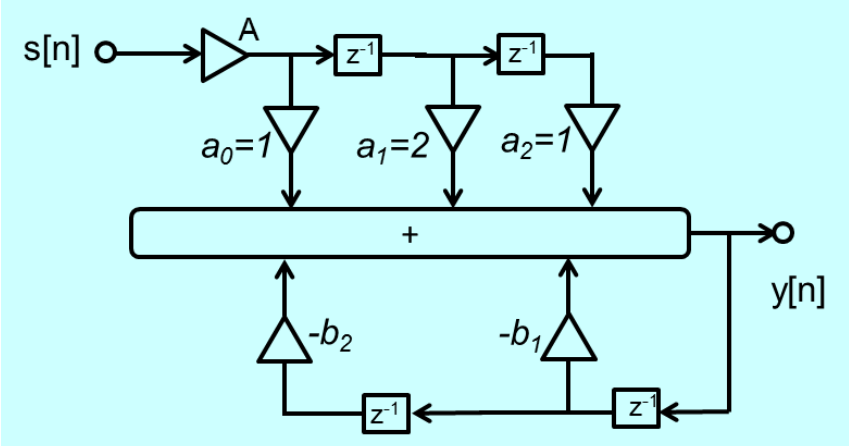
$$G(f) = G(z = e^{j\omega T_e})$$



$$\omega = 2\pi f \qquad \omega \in \left[-\frac{\pi}{T_e}, \frac{\pi}{T_e}\right]$$

Example (I)

How to obtain the frequency response from the transfer function G(z) ?



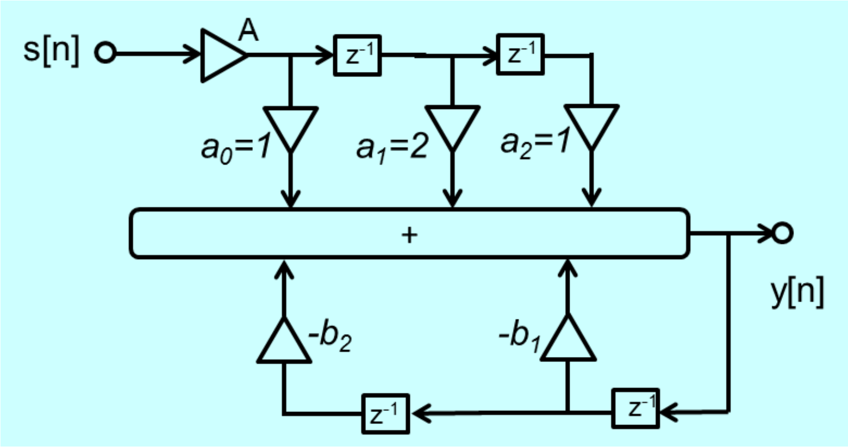
$$y[n] = A.\left(s[n] + 2.s[n-1] + s[n-2]\right) - b_1.y[n-1] - b_2.y[n-2]$$

Transfer function *G(z)* :

$$G(z) = \frac{Y(z)}{S(z)} = A\,\frac{1 + 2.\,z^{-1} + z^{-2}}{1 + b_1.\,z^{-1} + b_2.\,z^{-2}}$$

Example (II)

How to obtain the frequency response from the transfer function G(z) ?

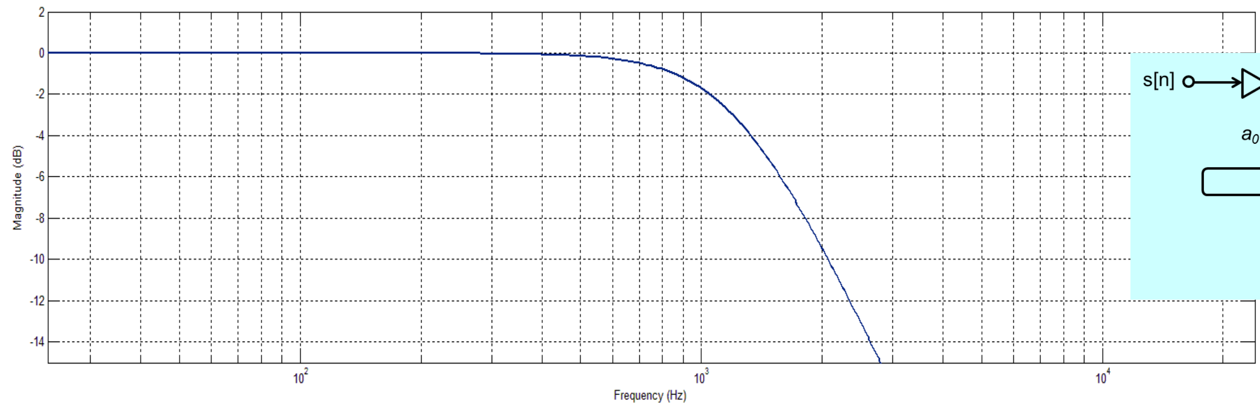


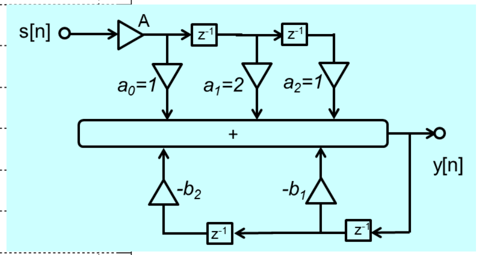
$$G(z)=\frac{Y(z)}{S(z)}=A\,\frac{1+2.\,z^{-1}+z^{-2}}{1+b_1.\,z^{-1}+b_2.\,z^{-2}} \qquad G(\omega)=G(z=e^{j\omega T_e})$$

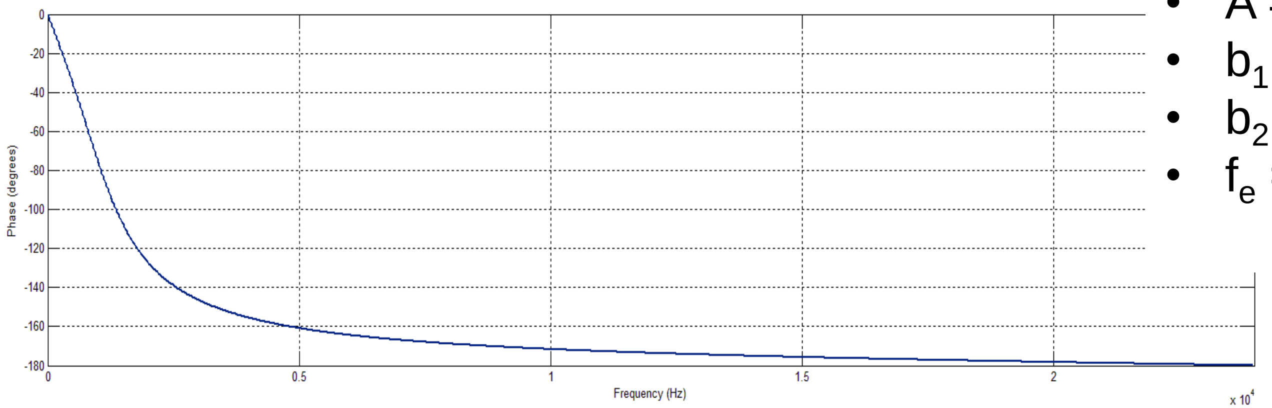
Matlab function: ***freqz(A,B,N,F$_e$)***

Example (III)

How to obtain the frequency response from the transfer function G(z) ?







- A = 0,005542
- b_1 = -1,7786,
- b_2 = -0,8008,
- f_e = 48 kHz.

Digital (audio) filters (VI)

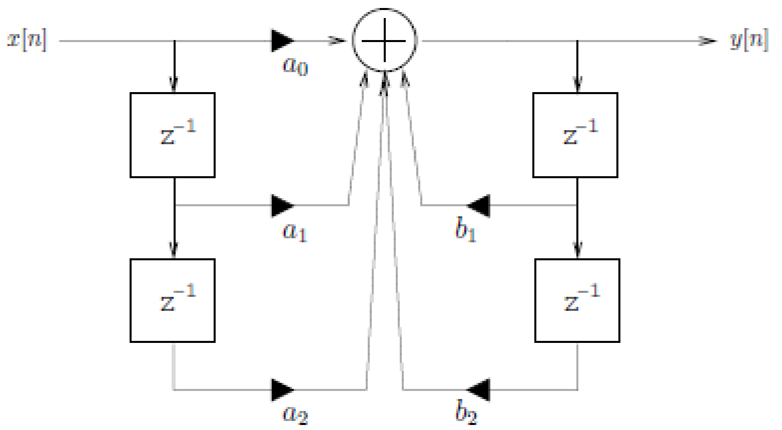


Figure 4.8: IIR Filter Direct Form I

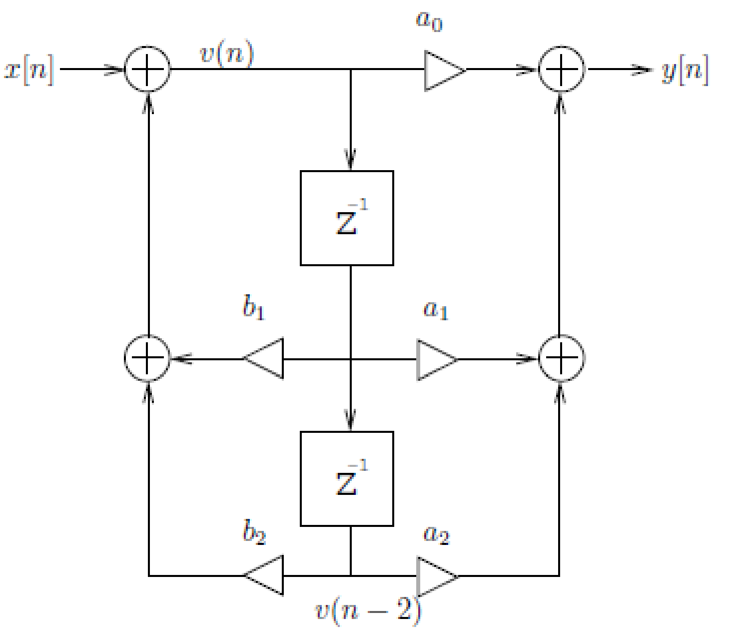


Figure 4.9: IIR Filter Direct Form II