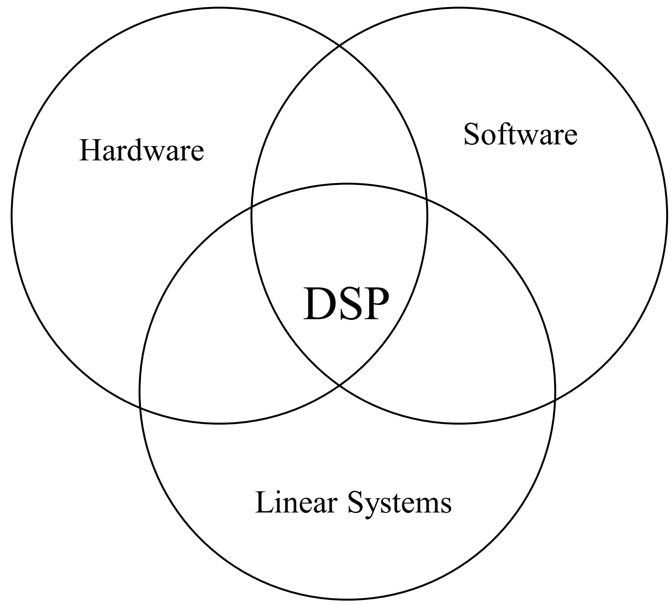
Course Notes 1 – Introduction

I. Overview of Digital Signal Processing
   A. What is Digital Signal Processing
   B. Signal Processing - Analog vs Digital
   C. Real-Time vs Non-Real Time DSP
   D. Digital Signal Processing Systems
   E. Useful Identities

A. What is Digital Signal Processing?



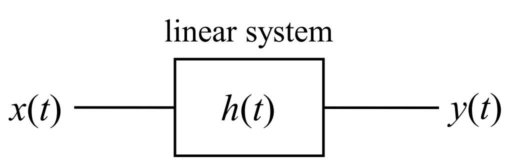
At the heart of Digital Signal Processing is *linear systems theory* which describes the signal processing algorithms in mathematical terms.

Hardware and software support the implementation of these algorithms in many different formats.

Digital Signal Processing is the representation of signals as a sequence of numbers and then processing or manipulating the sequence to achieve a desired goal.

- **computer workstation (software)**
- **specialized microprocessor (software)**
- **specialized digital devices (hardware)**
- **field programmable devices (software/hardware)**

You are already familiar with analog signal processing:



$h(t)$ = impulse response

The most fundamental description of the behavior of an analog circuit is the <u>differential equation</u>.

$$\sum_{k=0}^{N} a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} b_k \frac{d^k x(t)}{dt^k}$$

- Describes dynamic behavior of the system
- R, L, and C used to implement derivatives
- Diff Eqn gives little insight into the circuit implementation
- Easy analysis can be performed via Laplace transform

$$H(s) = \frac{\sum_{k=0}^{M} b_k s^k}{\sum_{k=0}^{N} a_k s^k}$$

where '*s*' is a complex variable in rectangular coordinates: $s = \sigma + j\Omega$

We can represent a linear system as:

- $h(t)$ - which is the response of the system to an impulse - the "impulse response"
- $H(s)$ - is traditionally named the "transfer function". From this expression we obtain the poles and zeros of the system
- $H(j\Omega)$ - is the frequency response. It can be obtained from the transfer function as described below

If we define

$$f\{\cdot\} = \text{the Fourier transform}$$

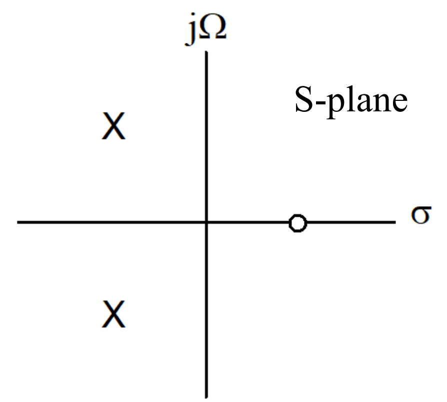
$$L\{\cdot\} = \text{the Laplace transform}$$

then

$$f\{h(t)\} = H(j\Omega)$$ the "frequency response"

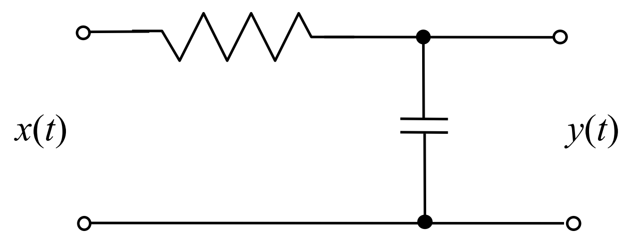
$$L\{h(t)\} = H(s)$$ the "transfer function"

$$H(s)|_{s=j\Omega} = H(j\Omega)$$ the frequency response is the transfer function evaluated on the jΩ axis



S-plane for continuous-tim LTI systems

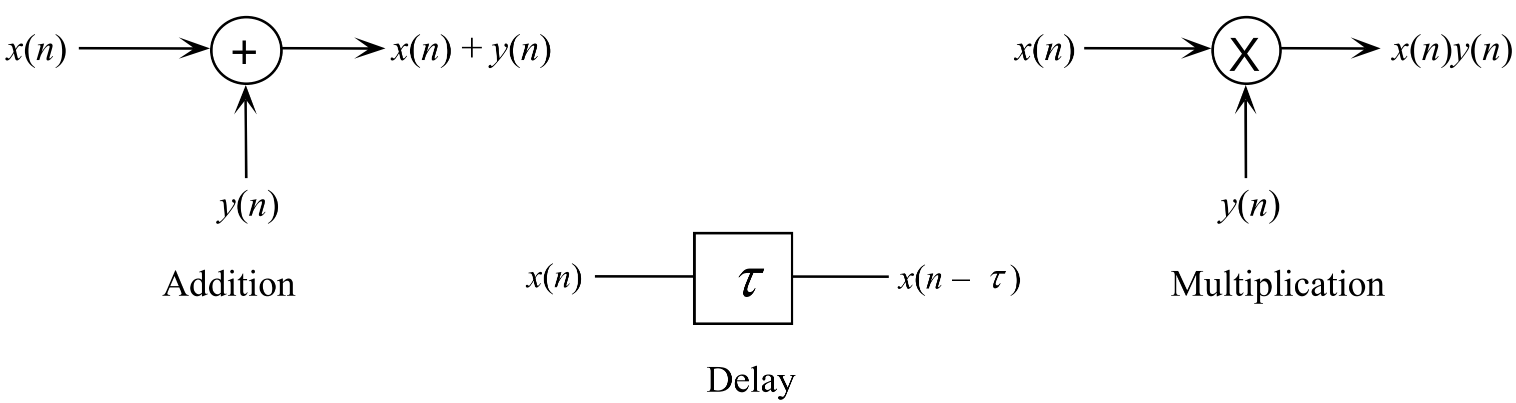
"Analog" or continuous-time systems are implemented using resistors, capacitors, inductors and active devices.



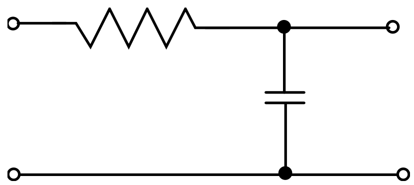
What is the function of this system?

The signals processed by systems of this type are continuous-time electrical currents or voltages having an amplitude which is continuous and infinitely variable.

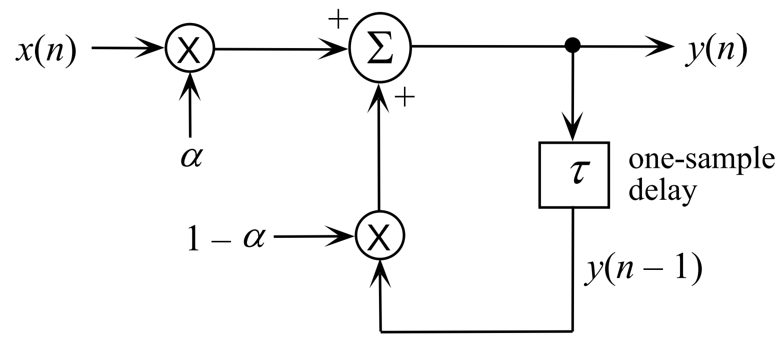
In <u>Digital Signal Processing</u>, we implement systems as mathematical operations, either in hardware or as software algorithms. All DSP algorithms can be implemented with only three operations:



For example, the previous lowpass filter can be implemented using DSP principles as shown below:

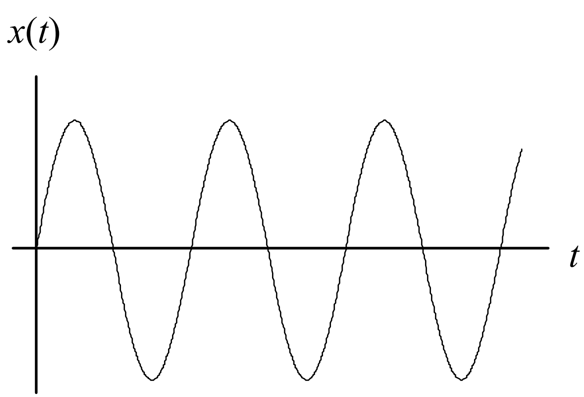


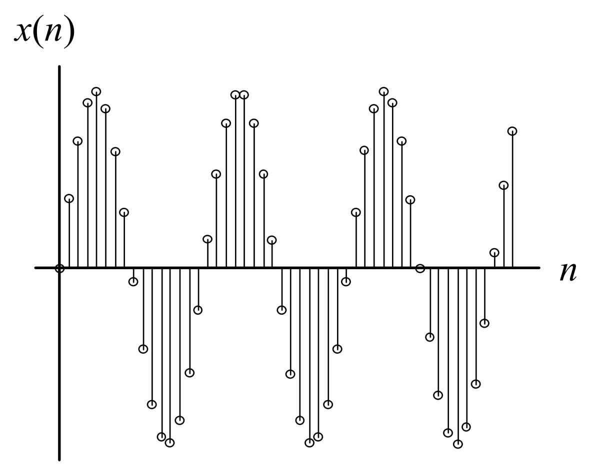
The analog implementation



The DSP implementation

Conceptually, we apply continuous-time (analog) signals to the analog system and discrete-time sequences to the DSP system.





B. Signal Processing - Analog versus Digital

LIMITATIONS OF ANALOG SIGNAL PROCESSING

Stability — Analog components are subject to drift (change in value) over time

Accuracy — Noise is inherent in all components. Therefore, a finite signal to noise ratio (SNR) exists in all analog systems

Flexibility — Analog systems are fixed for a particular design, and therefore are inflexible to design changes

Predictability — Analog systems have practical limitations on dynamic range. All production units are not identical.

ADVANTAGES OF DIGITAL SIGNAL PROCESSING

Stability — No drift in component values due to age, temperature, etc.

Accuracy — Determined by number of bits

Flexibility: — Many simultaneous tasks possible. System changed via software. Adaptability.

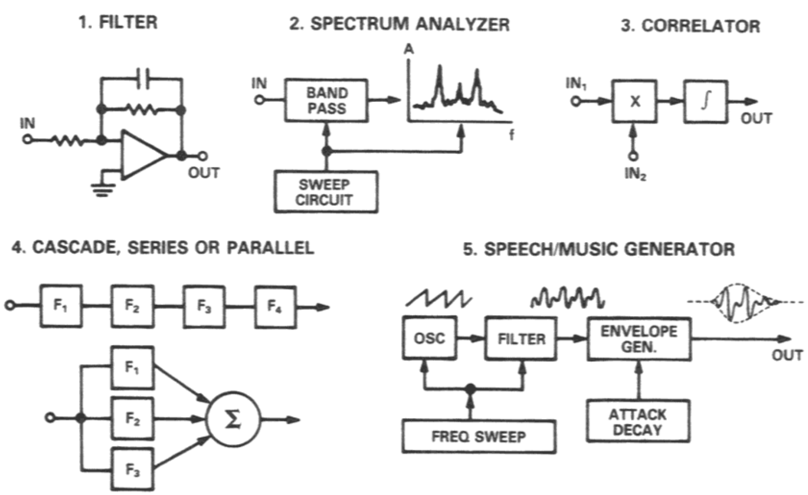
Predictability — Computer simulation is very effective. All production units are identical and no fine tuning is required

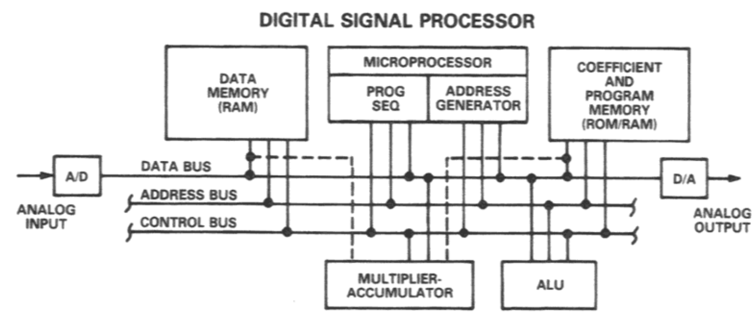
DISADVANTAGES OF DIGITAL SIGNAL PROCESSING

Cost — At the present time digital systems are often not as cheap as their analog counterparts, but prices are steadily falling.

Complexity — Requires both complex hardware and software.

Speed — Limited throughput is often the factor that influences the decision to not use DSP. Signals with large bandwidth require fast processors.

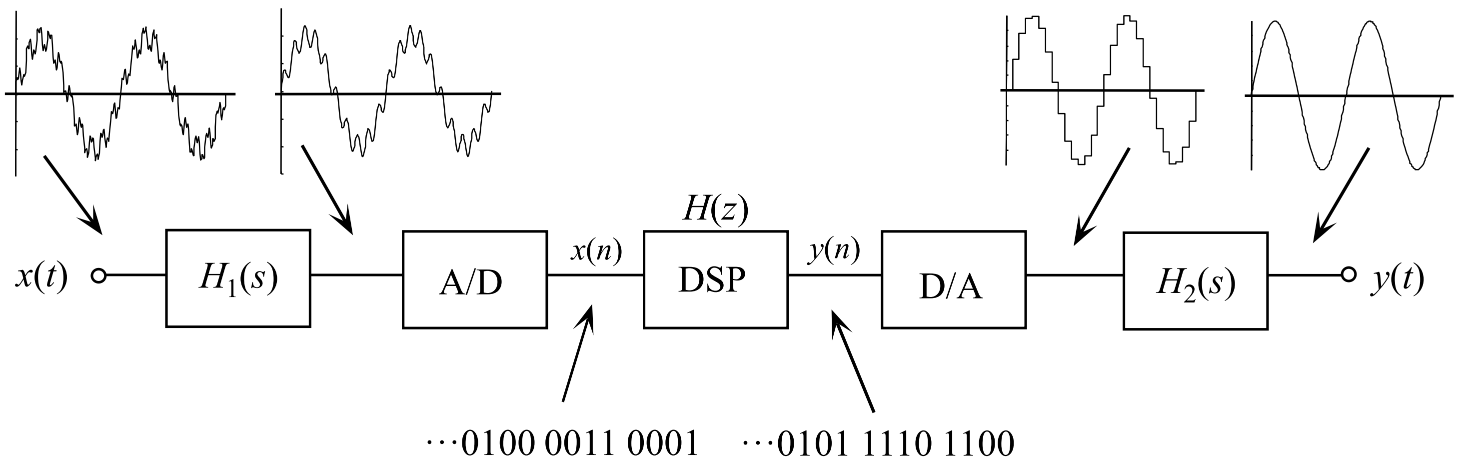




The figures above illustrate that while analog systems require a specific architecture (*i.e.*, circuit) in order to implement a specific function, digital processors can accomplish all of the functions with a single architecture.

C. Digital Signal Processing Systems

The general concept of DSP - especially as it applies to real-time applications - is shown below:



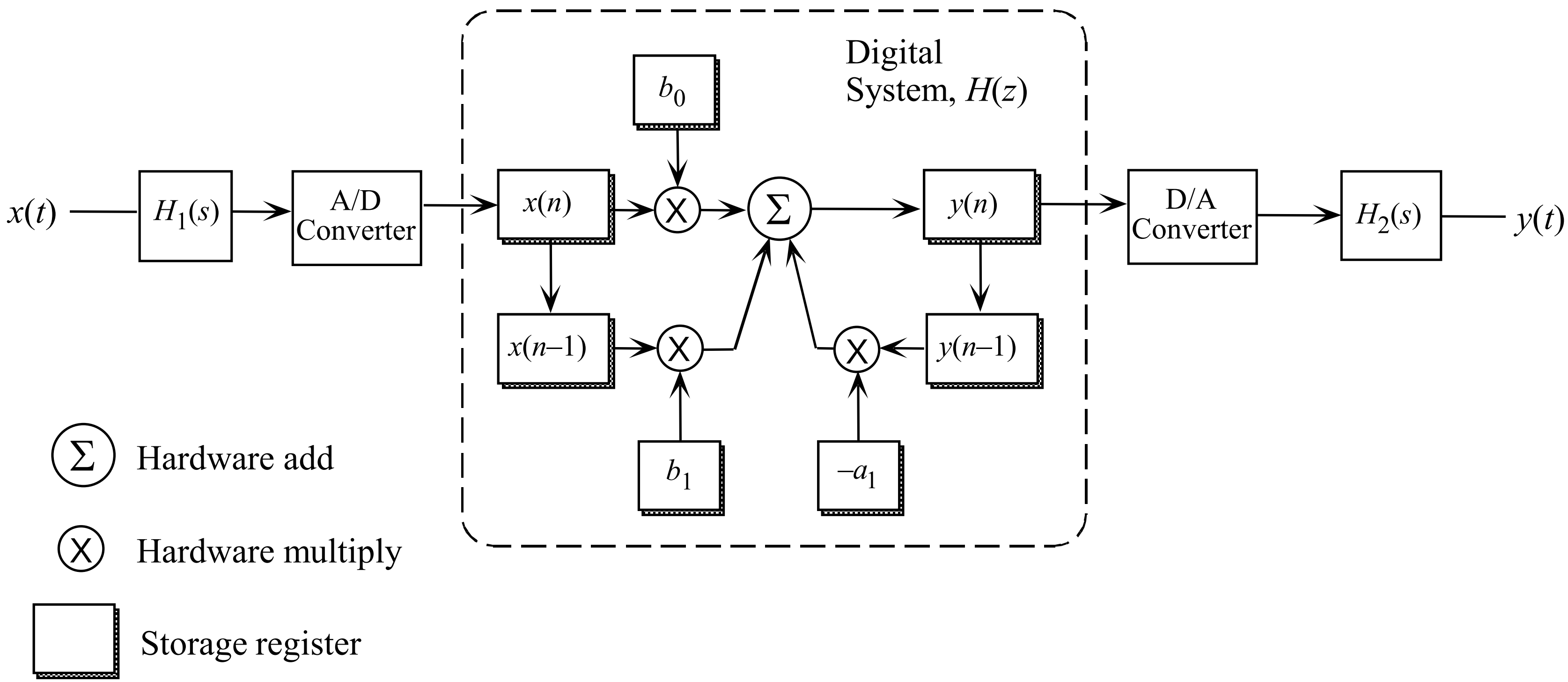
$H_1(s)$ - anti-aliasing analog filter
$H_2(s)$ - anti-imaging reconstruction filter
$H(z)$ - DSP system

Note: While this course will focus upon linear systems, *i.e.* like $H(z)$ above, DSP provides a means to also incorporate complex non-linear operations as well. In other words, DSP is much more than simply the discrete version of an analog circuit.

How will we construct $H(z)$? ...with a numerical algorithm. For example,

$$y(n) = b_0 x(n) + b_1 x(n-1) - a_1 y(n-1)$$

This algorithm can be implemented in hardware with the following digital system:



The algorithm that describes a discrete-time system is actually a difference equation, which is analogous in every way to a differential equation, which describes the behavior of an analog, continuous-time system.

$$\sum_{k=0}^{N} a_k\, y(n-k) = \sum_{k=0}^{M} b_k\, x(n-k)$$

- Describes dynamic behavior of the system
- Add, multiply and delay used to implement differencing operations
- The difference equation is the implementation of the system (*i.e.*, an algorithm)
- Easy analysis via *z*-transform

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}$$

where 'z' is a complex variable in polar coordinates:

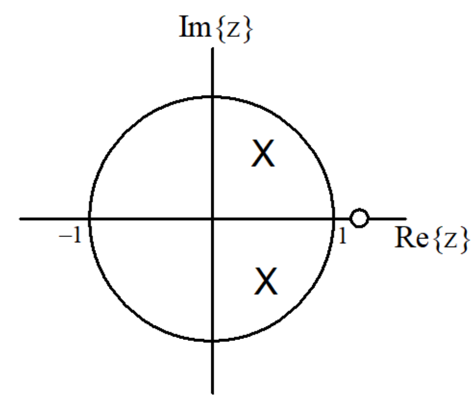
$$z = r\, e^{j\omega}$$

We can analyze the behavior of the discrete systems with mathematical tools that are analogous to our analog tools. For example,

- $h(n)$ - which is the response of the system to an impulse sequence - the "impulse response"

- $H(z)$ - is traditionally named the "system function". It is actually the discrete-time transfer function. From this expression we obtain the poles and zeros of the discrete-time system.

- $H(\omega)$ - is the frequency response. It can be obtained from the system function as described below



Z-plane for discrete-time LSI systems

If we define

$f\{\ \}$ = the Fourier transform

$Z\{\ \}$ = the z-transform

then

$Z\{h(n)\} = H(z)$ the "system function"

$f\{h(n)\} = H(\omega)$ the "frequency response"

$H(z)\big|_{z=e^{j\omega}} = H(\omega)$ the frequency response is the transfer function evaluated on the unit circle

(note: in many texts $H(e^{j\omega})$ is used instead of $H(\omega)$ to indicate frequency response)

In this course we will demonstrate that any linear shift invariant system can be implemented as an algorithm having the form of a linear constant coefficient difference equation:

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) - \sum_{k=1}^{N} a_k y(n-k)$$

From this algorithm we can obtain the system function $H(z)$ by using the z-transform:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

We will manipulate $H(z)$ and implement several types of finite impulse response (FIR) and infinite impulse response (IIR) systems.

As the name indicates, an IIR system has an impulse response which is infinite in extent. It is usually implemented as a recursive system. Conversely, a FIR system has a finite number of values in its impulse response.

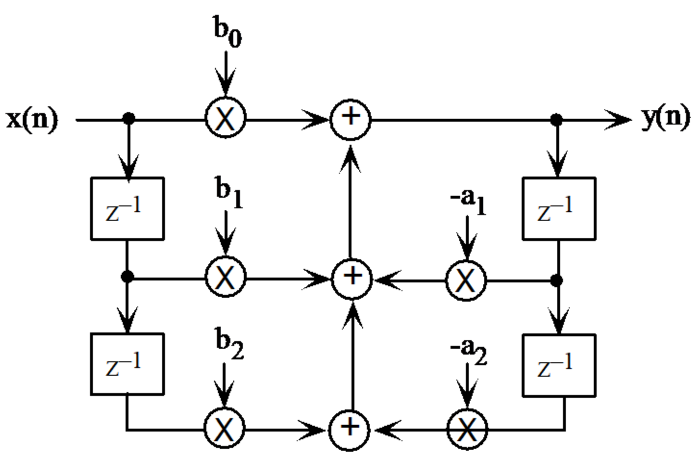
DSP ALGORITHM STRUCTURES

The DSP structure is analogous to a *schematic diagram* of the algorithm

Algorithms are classified into two broad categories of structures:

- finite impulse response (FIR) systems
- infinite impulse response (IIR) systems

As the name indicates, an IIR system has an impulse response which is infinite in extent. It is usually implemented as a recursive system (*i.e*. employs feedback). A FIR system has a finite number of values in its impulse response. The FIR system is usually implemented as a non-recursive feed-forward structure as shown on the next page.

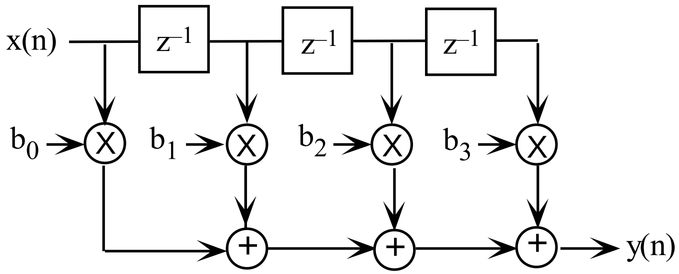


Infinite Impulse Response (IIR) Configuration

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$H(\omega) = \frac{b_0 + b_1 e^{-j\omega} + b_2 e^{-j2\omega}}{1 + a_1 e^{-j\omega} + a_2 e^{-j2\omega}}$$



Finite Impulse Response (FIR) Configuration

Difference equation (algorithm):

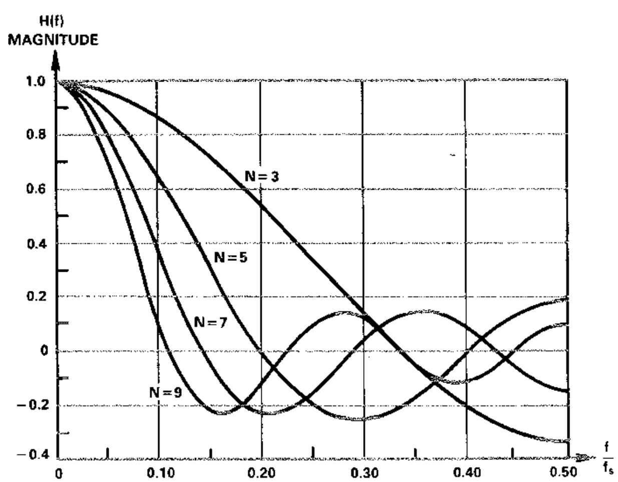
$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + b_3 x(n-3)$$

System Function:

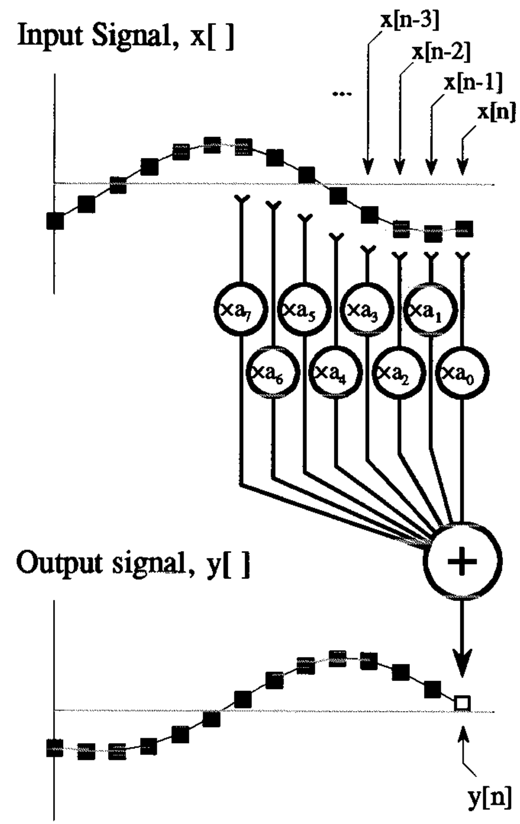
$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}$$

Frequency Response:

$$H(\omega) = b_0 + b_1 e^{-j\omega} + b_2 e^{-j2\omega} + b_3 e^{-j3\omega}$$



a. Moving-average filter—various numbers of samples.



$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + a_3 x[n-3] + a_4 x[n-4] + \cdots$$

D. Real-Time versus Non-Real-Time Digital Signal Processing

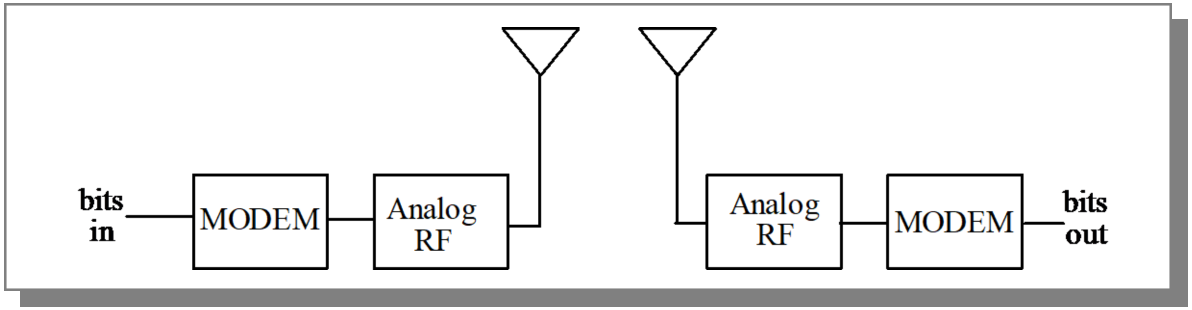
Real-Time DSP

- Real-Time DSP in its simplest form is analog processing done digitally
- The computations keep pace with the input and output signals

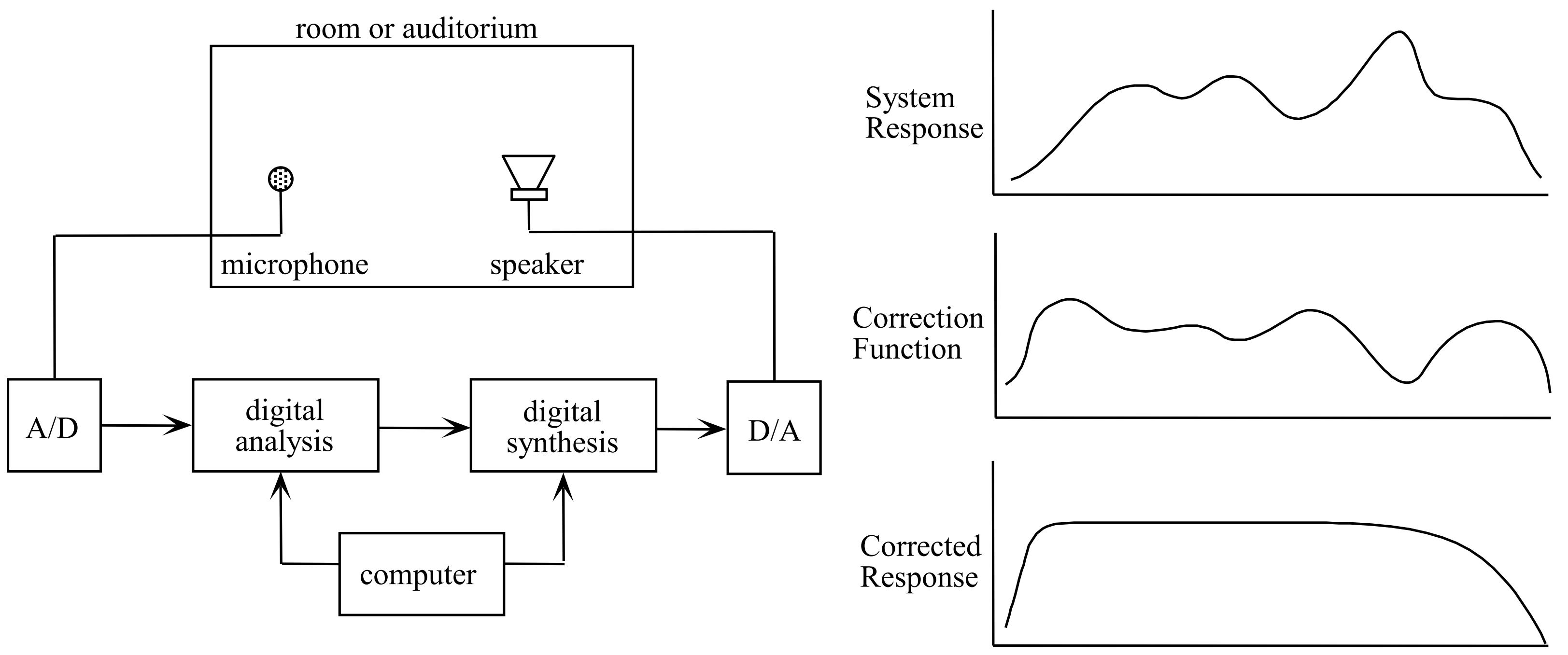
The systems that perform real-time DSP are usually some type of **filter**. Some applications are:

- removal of unwanted background noise
- removal of interference
- separation of frequency bands
- shaping of signal spectrum

An important contemporary real-time application is wireless communications. The system must be able to provide a specified number of bits per second.

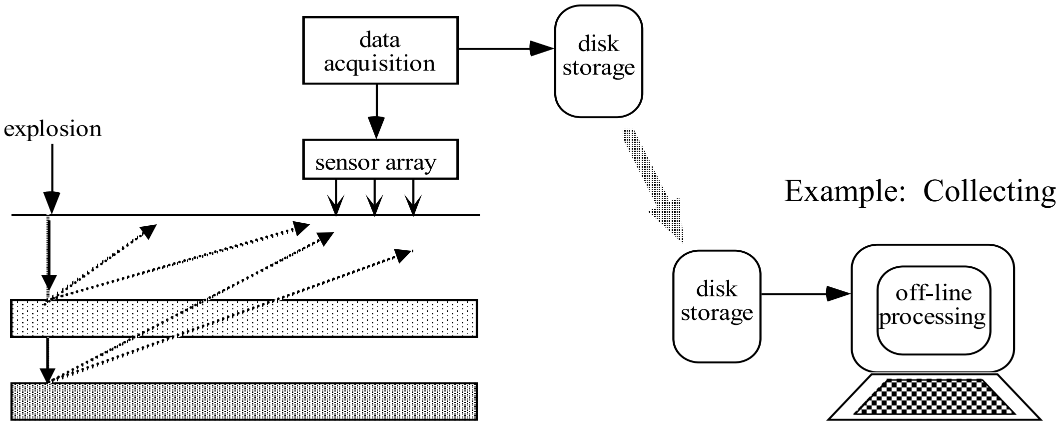


An example of Real-Time digital signal processing is shown below: A DSP system is used to equalize a room or theater for optimum acoustical effect



Non-Real-Time DSP

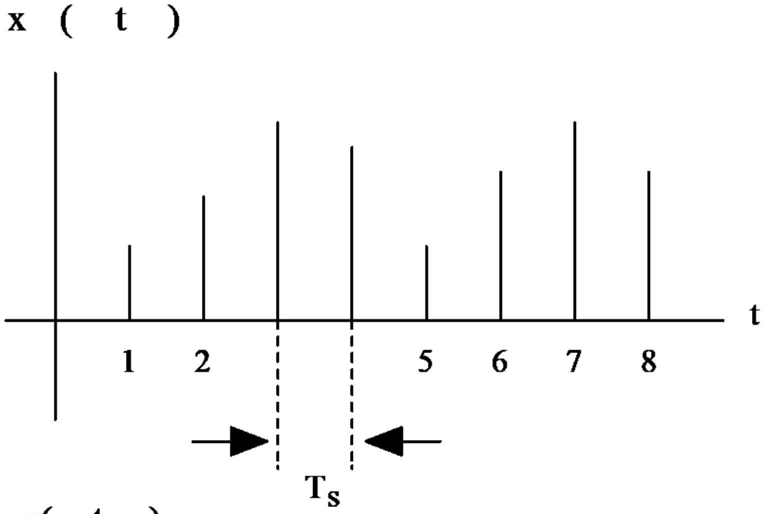
- All input data is available
- Processing time is not critical
- Usually applied to complex signal analysis tasks
- Suitable for computationally intensive techniques (*e.g.*, time-frequency transformations)
- DSP Algorithms usually implemented on a general purpose processor

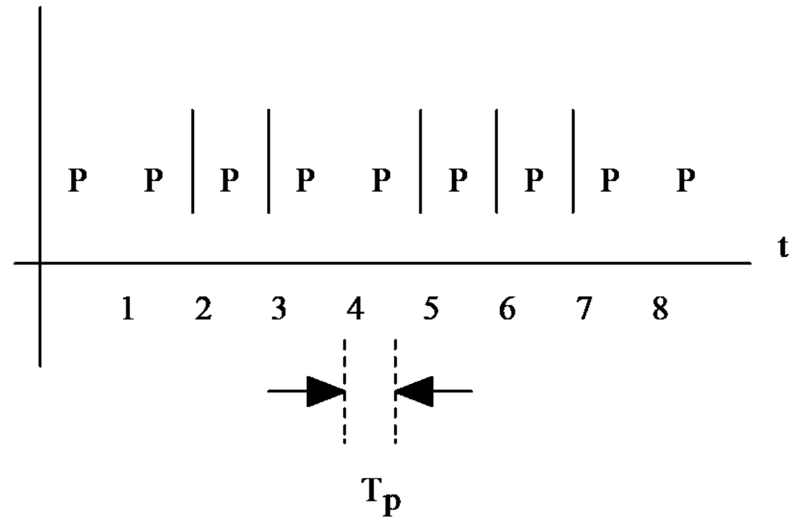


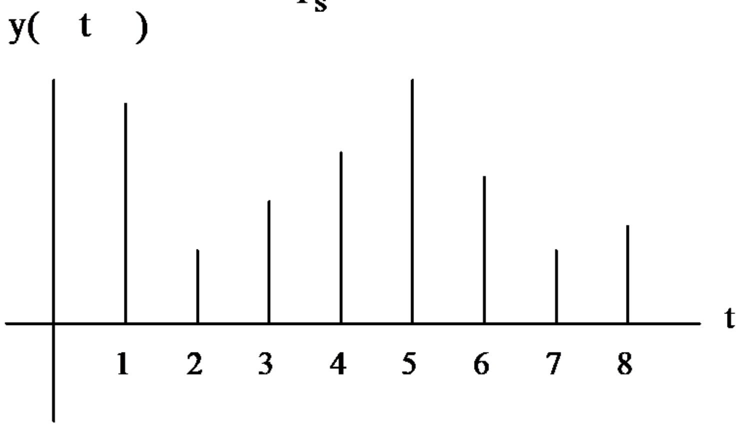
Example: Collecting seismic data

DSP Data Handling Modes

1. Stream Processing - The overriding goal is to move data in, perform the math and move the data out before the next sample is available

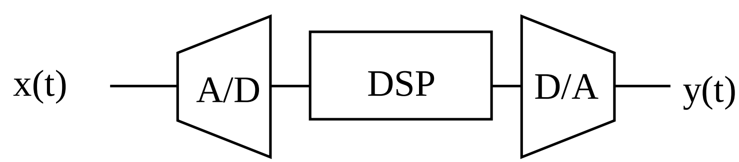




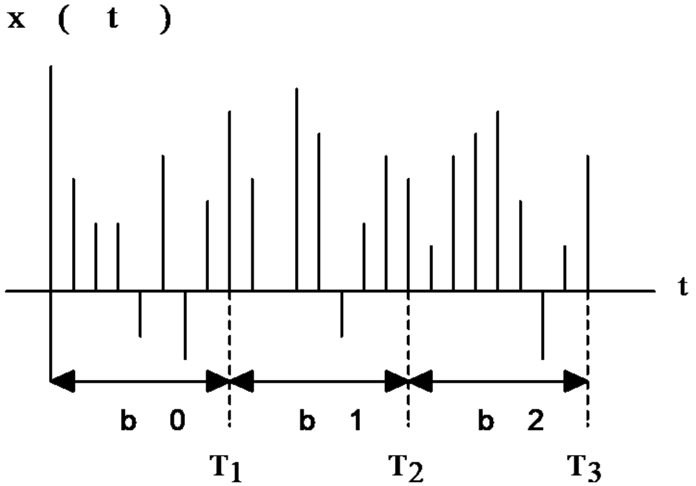


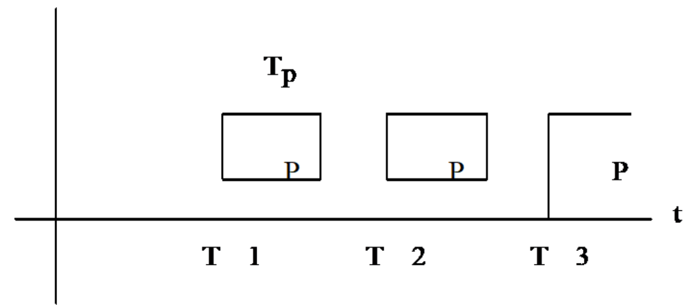
Note that Tp is the processing time.

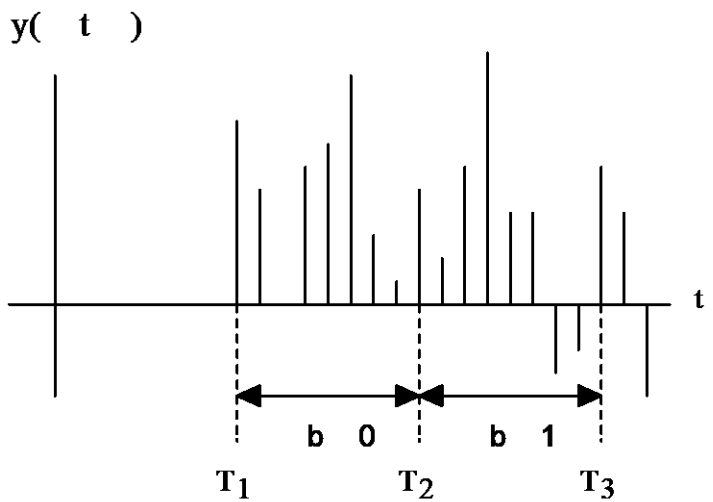
For this mode we require Tp < Ts



2. Block Processing - The overriding goal is to move a block of data in, perform the math and move the data out before the next block of samples is available.







The data block size is b_k and $b_k = NT_S$. The processing time for each block is Tp.

For this mode we require that $Tp < NTs$

E. Useful Identities

COMPLEX NUMBERS

Let $j = \sqrt{-1}$, then $z = x + j\,y$.

$x = \mathrm{Re}\{z\}$ = real part, $y = \mathrm{Im}\{z\}$ = imaginary part.

COMPLEX OPERATIONS

$z_1 + z_2 = (x_1 + x_2) + j(y_1 + y_2)$: Complex addition

$z_1 \cdot z_2 = x_1 \cdot x_2 - y_1 \cdot y_2 + j(x_1 \cdot y_2 + x_2 \cdot y_1)$: Complex multiplication

$$\frac{z_1}{z_2} = \frac{x_1 \cdot x_2 + y_1 \cdot y_2}{x_2^2 + y_2^2} + \frac{x_2 \cdot y_1 - x_1 \cdot y_2}{x_2^2 + y_2^2}$$

: Complex division

$z^* = x - j\,y$: Complex conjugation

POLAR NOTATION

$z = |z| e^{j \arg\{z\}}$ where $|z|$ = magnitude, and $\arg\{z\}$ = phase = φ

$$z_1 \cdot z_2 = |z_1| \cdot |z_2| e^{j(\arg\{z_1\} + \arg\{z_2\})}$$: Polar multiplication

$$\frac{z_1}{z_2} = \frac{|z_1|}{|z_2|} e^{j(\arg\{z_1\} - \arg\{z_2\})}$$: Polar division

$$z^* = |z| e^{-j \arg\{z\}}$$: Polar complex conjugation

EULER'S IDENTITY

$e^{j\phi} = \cos\phi + j \sin\phi$, or equivalently

$$\cos\phi = \frac{1}{2}\left(e^{j\phi} + e^{-j\phi}\right), \quad \sin\phi = \frac{1}{2j}\left(e^{j\phi} - e^{-j\phi}\right)$$

Hence, $e^{j0} = 1$, $e^{j(\pi/2)} = j$, $e^{j\pi} = -1$, $e^{j(3\pi/2)} = -j$, and $e^{j(2\pi)} = 1$

SUMMATION FORMULAE

1st term → r^k; # of terms → $N-k$

$$\sum_{i=k}^{N-1} r^i = \frac{r^k\left(1-r^{N-k}\right)}{1-r}$$

: sum of geometric series

1st + last → $(k+N-1)$; # of terms → $(N-k)$

$$\sum_{i=k}^{N-1} i = \frac{(k+N-1)(N-k)}{2}$$

: sum of arithmetic series

For example, compute

$$S = \sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}nk}$$

for n any integer