## 1.1 Introduction

**Basic idea:** communications various forms of information between people or (analog or digital) systems.

**Cartoon communications:**

```
```

Transmission line can be space, time, or neither. Typically we need to process the signal to make it suitable for the next stage (or to enhance the information for the receiver).

**Examples:**

<table>
<thead>
<tr>
<th>Source</th>
<th>Signal representation</th>
<th>Transmission line</th>
<th>Signal reconstruction</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Airplane</td>
<td>Electro-magnetic radio waves</td>
<td>Airwaves</td>
<td>Analog processing</td>
<td>Radar screen</td>
</tr>
<tr>
<td>2. Musical group</td>
<td>A/D converter and digital PCM</td>
<td>CD disk</td>
<td>D/A converter</td>
<td>Human ear</td>
</tr>
<tr>
<td>3. Disk drive</td>
<td>Digital PCM</td>
<td>Computer bus, CPU, memory</td>
<td>Digital PCM</td>
<td>Disk drive</td>
</tr>
<tr>
<td>4. Napster</td>
<td>MP3 file</td>
<td>The internet</td>
<td>Digital PCM and D/A</td>
<td>Computer speakers</td>
</tr>
</tbody>
</table>

In many cases we want to process (modify) the signal in complex ways. Can process this using analog circuitry (simple example, RC-circuits) or digital.

**Key point:** In many cases, there is no actual penalty to process things digitally (i.e., essentially perfect reconstruction of signals represented digitally). And it is sometimes better to do things digitally.

**Discrete/continuous time (discrete-time signal processing)**

**Discrete/continuous amplitude (quantization)**

Digital signals means both. Main focus of book is the first case. Later case is dealt with in vector quantization courses.

Digital signals come from:

1. Sampled analog signals
2. Computer generated signals

**Processing of discrete time data:** most often:

1. General purpose computer
2. Special purpose DSP chip
There are many hardware devices that can perform computations for DSP

1. DSP chips, e.g., TI TMS320C 3x, 4x, 5x, ..., e.g., Motorola DSP560xx, DSP96002 (see the dsp FAQ on the course web page for a long listing)

2. General purpose microprocessors
   ex.
   Intel Pentium III with SSE
   IBM RS6000 with parallel multiply-accumulate instructions
   Power PC

What are the main differences between the two types of processors?
flexibility
power consumption
floating point / fixed point
support for multi-user OS

Applications of DSP

- Military
  1. Sonar
  2. Radar
  3. battlefield acoustics

- Scientific
  1. seismic data (first industry to go digital)
  2. signal detection (SETI, project Phoenix)
  3. speech enhancement for handicapped

- Industrial
  1. Monitoring manufacturing processing via acoustic emissions

- Commercial/Entertainment
  1. CD/DVD players
  2. mpeg, mp3, music, sound, and video compression.
  3. Napster, etc.
  4. Speech recognition
  5. communications (phone)

History of DSP

1960s:
- numerical analysis (finite difference), too difficult to solve differential equation in continuous variable, use discrete “simulation”.
- simulating analog systems (because expensive to build otherwise)
- speech processing

limitations
need lots of storage for big data sets

advantages

flexible (algorithm design, can’t burn circuits, can implement algorithms that are not easily implemented in analog (FIR)).

DSP systems are now easy to design test:

1. matlab lots of tools to facilitate digital system design (DEMO MATLAB)
2. lots of existing code (e.g., FFTW, fastest fft available www.fftw.org)
3. relatively easy to implement new systems in C/C++/Fortran or even Java

**Journals for DSP** (see www.ieee.org web page)

- IEEE transaction on signal processing
- IEEE signal processing letter
- IEEE transactions on speech and audio processing
- IEEE signal processing magazine

The above journals are free if access online from University of Washington campus through ieeexplore.ieee.org.

This class will be mostly theory and practice of general DSP techniques, without singling out any particular application. EE519 is more of an applications class.

There are also advanced DSP courses that go in to adaptative and array signal processing, various forms of FFT, wavelets, stochastic processes. There will also be specific courses for speech processing (EE516) (next spring there will be a two quarter course on speech and language signal processing). EE519

### 1.2 Notation for Discrete Sequences

\( \{x\} \), a set of numbers = \( \{x[n]\} \) \(-\infty < n < \infty, n = \text{integer}, \)

could arise from periodic sampling of analog signal

\[
x[n] = x_a(nT) \quad T = \text{period} = \frac{1}{f}, \quad f = \text{frequency}
\]

**Ex:** \( x[n] \)

\[
x[n] = \begin{cases} 1 & 0 \leq n \leq 3 \\ 0 & \text{else} \end{cases}
\]

![Figure 1.1: x[n]](image-url)
Discete:

\[ x[n] = A \cos(\omega_0 n + \phi) \quad x_a(t) = A \cos(\Omega t + \phi) \]

where \( A \) is amplitude, \( \omega_0 \) is frequency in radians/sample, \( n \) is dimensionless in unit 1 or samples, \( \phi \) is phase in radians, \( \Omega \) is frequency in radians/sec, \( t \) is time in seconds.

**Ex:** sequences we will see lots of

1. Delayed Sequence

\[
\{y[n]\} = \{x[n-n_0]\}
\]

2. Unit Sample Sequence

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

3. Unit Step Sequence

\[
u[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n-k]
\]

when \( n-k = 0 \) \( \delta[n-k] = 1 \) \( k = n \)

4. Unit Difference Sequence

\[
\delta^{(1)}[n] = \begin{cases} 
1 & n = 0 \\
-1 & n = 1 \\
0 & \text{else}
\end{cases}
\]

\[
= \delta[n] - \delta[n-1] \quad \text{(like a 1st order difference)}
\]

\[
= u[n] - 2u[n-1] + u[n-2] \quad \text{(like a 2nd order difference)}
\]
5. Complex Sinusoid

\[ x[n] = A \alpha^n \quad (\alpha = |\alpha| e^{j\omega_0}) \]

\[ = A |\alpha|^n e^{j\omega_0 n} \quad \text{(damped sampled exponential)} \]

\[ = |A| |\alpha|^n e^{j\omega_0 n + \phi} \quad (A = |A| e^{j\phi}) \]

\[ = |A| |\alpha|^n (\cos(\omega_0 n + \phi) + j \sin(\omega_0 n + \phi)) \quad \text{(by Euler’s equation)} \]

where \( A \) is amplitude, \( \alpha \) is geometric growth, \( e^{j\omega_0 n} \) is sinusoidal modulation

6. Periodic Sequence

\[ x[n] = x[n + N] \quad \forall n \text{ and some integer } N, N \in \mathbb{Z} \]

**Definition 1.1 (Continuous Time Periodicity).**

\[ \exists T \text{ s.t. } x(t) = x(t + nT) \quad \forall t \]

**Question 1:** Is \( A \cos(\omega_0 t + \phi) \) periodic \( \forall \omega_0 \)? Yes.

**Question 2:** Is \( x[n] = A \cos(\omega_0 n + \phi) \) periodic \( \forall \omega_0 \)? No.

Not \( \forall \omega_0 \), conditions.

\[ \cos(\omega_0 (n + N) + \phi) = \cos(\omega_0 n + \phi) \]

So \( \omega_0 N \) must be a multiple of \( 2\pi \), i.e., \( \omega_0 N = k \cdot 2\pi \) or \( \omega_0 = \frac{2\pi k}{N} \)

### 1.3 Discrete Time Systems

**Definition 1.2 (Discrete Time Systems).** A discrete time system maps the input sequence, \( x[n] \), to a new sequence, the output sequence \( y[n] \).

![Figure 1.5: Discrete Time Systems](image)

We will study properties of discrete time input signals and systems.

**Ex:** Delay Systems

\[ y[n] = x[n - n_d] \quad -\infty < n < \infty \]

where \( n_d \) is the delay.
Ex: Moving Average Systems

\[ y[n] = \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} x[n-k] \]

Definition 1.3 (Memoryless Systems). Values of \( y[n] \) depends only on \( x[n] \).

Ex:

\[ y[n] = \log x[n] \]

Definition 1.4 (Linear Systems, i.e., Superposition). \( T\{\cdot\} \) is linear iff:

when \( y_1 = T\{x_1\} \) and \( y_2 = T\{x_2\} \),

then:

(additive property) \[ T\{x_1 + x_2\} = T\{x_1\} + T\{x_2\} = y_1 + y_2 \]

and

(scaling property) \[ T\{ax_1\} = aT\{x_1\} \]

or

(superposition) \[ T\{ax_1 + bx_2\} = aT\{x_1\} + bT\{x_2\} \]

Ex: Is \( y[n] = \log x[n] \) linear?

No. Counter example.
say \( x_1[0] = 0, x_2[0] = 1 \),

\[ \log(x_1[0] + x_2[0]) = 0 \neq -\infty + 0 \]

Ex: Is \( y[n] = \begin{cases} x[n] & n \geq 0 \\ -x[n] & n < 0 \end{cases} \) linear?

Yes.

\[ T\{ax_1 + bx_2\} = \begin{cases} ax_1 + bx_2 & n \geq 0 \\ -(ax_1 + bx_2) & n < 0 \end{cases} \]

\[ = a \begin{cases} x_1 & n \geq 0 \\ -x_1 & n < 0 \end{cases} + b \begin{cases} x_2 & n \geq 0 \\ -x_2 & n < 0 \end{cases} \]

Ex: Is accumulation \( y[n] = \sum_{k=-\infty}^{n} x[k] \) linear?

Yes.

\[ \sum_{k=-\infty}^{n} ax_1 + bx_2 = a \sum_{k=-\infty}^{n} x_1 + b \sum_{k=-\infty}^{n} x_2 = ay_1[n] + by_2[n] \]

Definition 1.5 (Time Invariance or Shift Invariance). Shifting the input produces an output that is equivalently shifted, i.e., if \( y[n] = T\{x[n]\} \) then \( T\{x[n-N]\} = y[n-N] \).

Definition 1.6 (Causality). A system does not respond until after the input starts, i.e., if \( x[n] = 0 \) for \( n < n_0 \) then \( y[n] = 0 \) for \( n < n_0 \). (Output can not anticipate input.)

Definition 1.7 (Bounded Input/Bounded Output (BIBO) or Stable). If \( |x[n]| \leq B_x < \infty, \forall n \), then \( |y[n]| \leq B_y < \infty, \forall n \).
\[ y[n] = \frac{1}{3} \sum_{k=-1}^{1} x[n-k] \] (See Fig. 1.7)

Ex: \( y[n] = \frac{1}{3} \sum_{k=-1}^{1} x[n-k] \) (See Fig. 1.7)


Ex:

\[ y[n] = \frac{1}{N} \sum_{k=n-(N-1)}^{n} x[k] \]

i.e., average up previous \( N \) samples

\( N = 1 \Rightarrow y[n] = x[n] \) identity system
\( N = 2 \Rightarrow y[n] = \frac{1}{2}(x[n] + x[n-1]) \)
\( N = 3 \Rightarrow y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2]) \)


### 1.4 Demonstration of MATLAB

Choose a volunteer, to record something (4 seconds of speech): (default 8KHz sampling rate)

```matlab
y = autorecorder; % creates a record object
recordblocking(y,4); % records 4 seconds of speech
% note, other way of recording is to
% use waverecord, previous version of matlab

1. Get the array of a signal value
```
s = getaudiodata(y);

2. Playback
   sound(s);

3. Play back in time reverse
   sound(s(length(s):-1:1));

4. Plot the signal
   plot(s);

5. Listen to negative of signal
   sound(-s);

   Q: why is it the same? A: just a uniform phase shift, we’ll learn why in this course.

6. Multiplied by a constant
   sound(s/2);

7. Square of the signal (make a little louder)
   sound(3 * s^.2)

   non-linear distortion.

8. Cube root of the signal (make a little soft)
   sound(cbrt(s)/3);

9. Spectral content of signal
   freqz(s, [1], [], 8000);

10. *Linear* frequency shift of signal, simple MATLAB routine that I wrote.

   ss = freqshift(s, 8000, 5000);
   sound(ss);
   freqz(ss, [1], [], 8000);

11. Spectrogram of the signal
    specgram(s, 512, 8000, [], 512-32);

12. Spectrogram of frequency shifted signal
    specgram(ss, 512, 8000, [], 512-32);

13. Design a simple band pass filter, passing only the components from 1k to 2kHz.

    filt = kaiserwinbp(2*pi*1/8,2*pi*1.01/8,2*pi*2/8,2*pi*2.1/8,0.001);
    sf = filter(filt,[1],s);
    specgram(sf,512,8000,[],512-32);
    sound(sf)
14. How about from .5k to 1kHz?

```matlab
filt = kaiserwinbp(2*pi*.5/8, 2*pi*.51/8, 2*pi*1/8, 2*pi*1.01/8, 0.001);
sf = filter(filt, [1], s);
specgram(sf, 512, 8000, [], 512-32);
sound(sf)
```

15. Make it twice as fast:

```matlab
sound(s(1:2:length(s)));
```

But there are problems with this, aliasing.

And so on.