ECE 246/446: Digital Signal Processing

http://www.ece.rochester.edu/courses/ECE446

Fall 2004

Introduction

- The aims of this lecture are:
  - To introduce the basic concept of signal processing
  - To explain the typical structure of a DSP system
  - To explain the benefits of DSP
  - To introduce some applications of DSP
  - To explain different architectures/types of DSPs

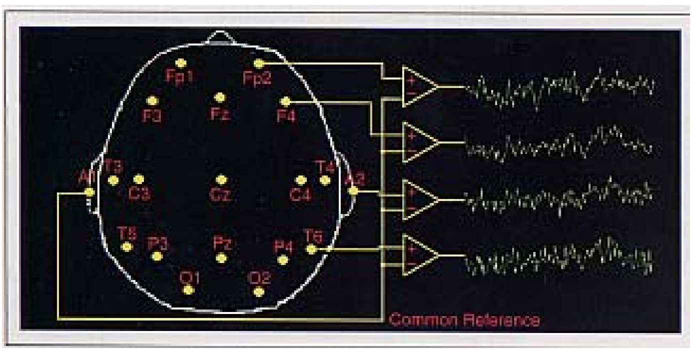
What is a signal?

- A function of independent variables such as time, distance, position, temperature, pressure, etc.
- A signal carries information
- Examples: speech, music, seismic, image and video
- A signal can be a function of one, two or *N* independent variables
  - Speech is a 1-D signal as a function of time
  - An image is a 2-D signal as a function of space
  - Video is a 3-D signal as a function of space and time

More Example Signals

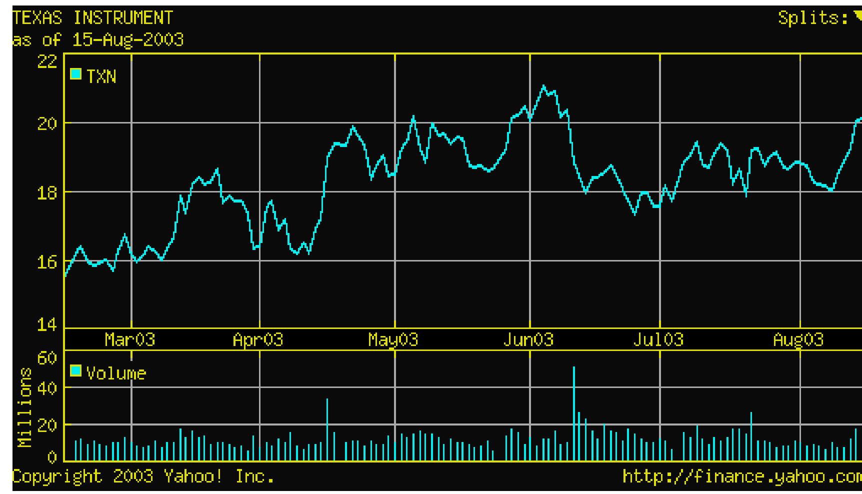
EEG

position

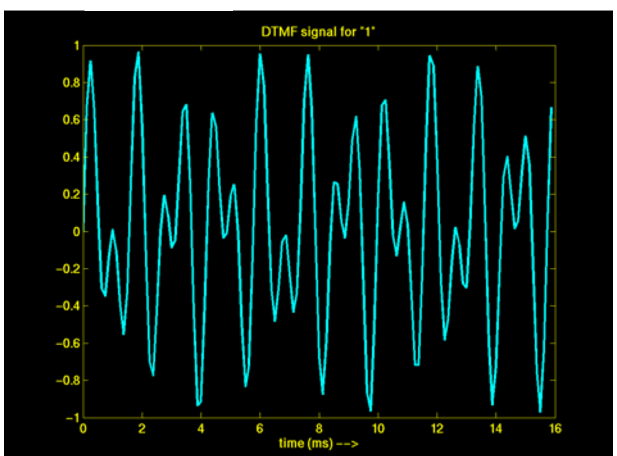


time

Stock price & volume



DTMF



time

Video

time

Types of Signals

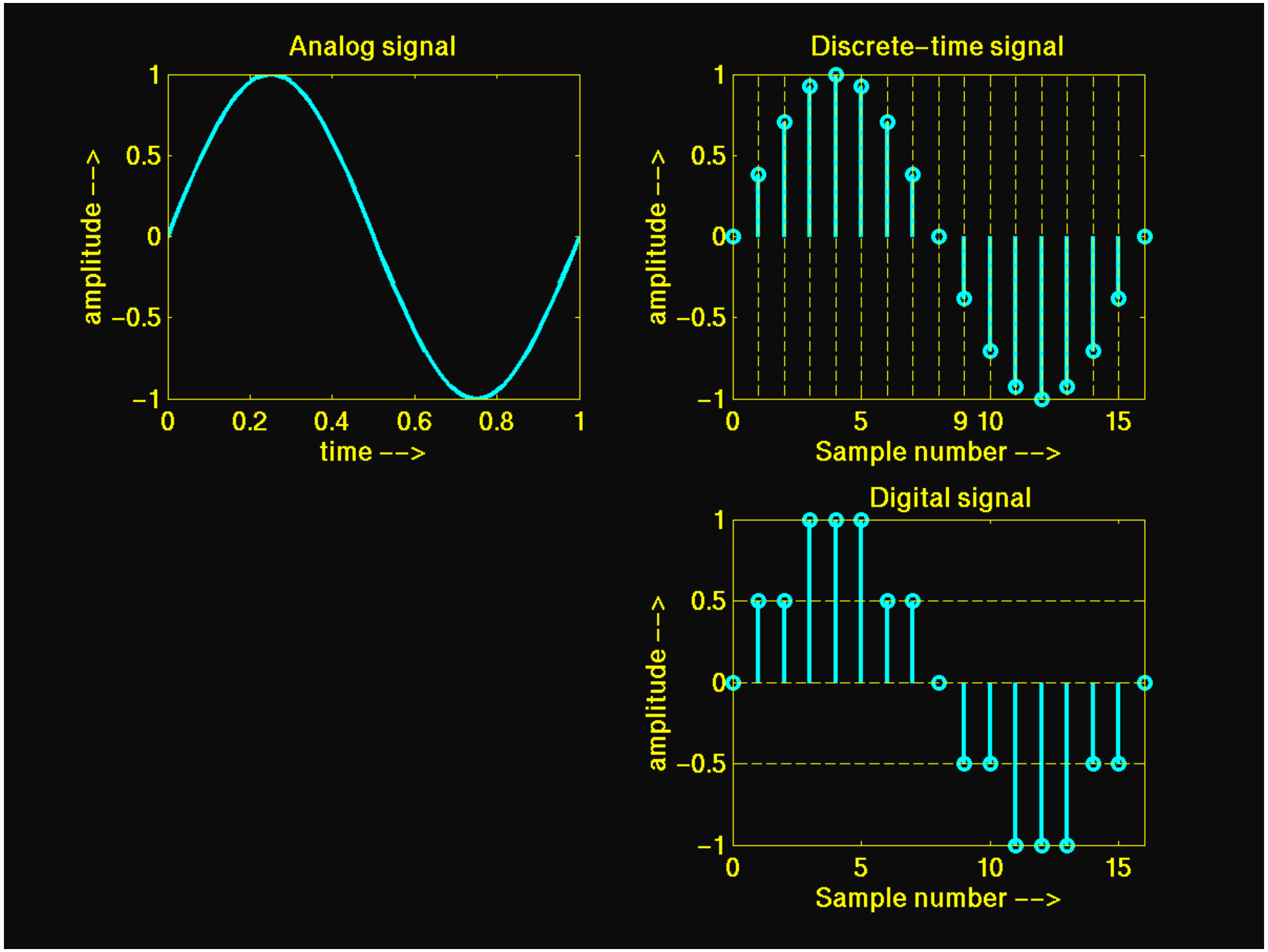
- **Analog Signals** (Continuous-Time Signals)
  Signals that are continuous in both the dependant and independent variable (e.g., amplitude and time). Most environmental signals are continuous-time signals.

- **Discrete Sequences** (Discrete-Time Signals)
  Signals that are continuous in the dependant variable (e.g., amplitude) but discrete in the independent variable (e.g., time). They are typically associated with sampling of continuous-time signals.

Types of Signals (cont.)

- **Digital Signals**
  Signals that are discrete in both the dependant and independent variable (e.g., amplitude and time) are digital signals. These are created by quantizing and sampling continuous-time signals or as data signals (e.g., stock market price fluctuations).

Types of Signals (cont.)



What is DSP?

- Changing or analyzing information that is measured as discrete sequences of numbers
- The representation, transformation, and manipulation of signals and the information they contain

Unique Features of DSP

- Signals come from the real world
  - Need to react in real time
  - Need to measure signals and convert them to digital numbers
- Signals are discrete
  - Information in between discrete samples is lost

Processing Real Signals

- Most of the signals in our environment are analog such as sound, temperature and light
- To processes these signals with a computer, we must:

  1. **convert the analog signals into electrical signals**, e.g., using a transducer such as a microphone to convert sound into electrical signal

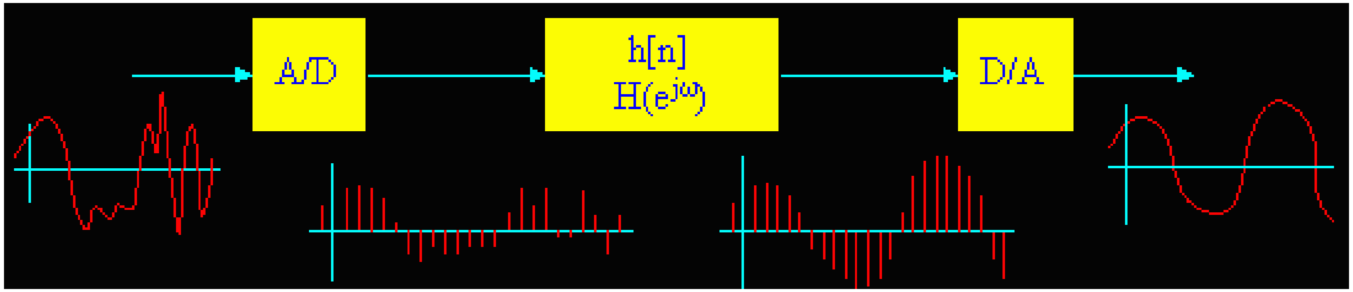
  2. **digitize these signals**, or convert them from analog to digital, using an ADC (Analog to Digital Converter)

Processing Real Signals (cont.)

- In digital form, signal can be manipulated
- Processed signal may need to be converted back to an analog signal before being passed to an actuator (e.g., a loudspeaker)
  - Digital to analog conversion can be done by a DAC (Digital to Analog Converter)

Typical DSP System Components

- Input lowpass filter (anti-aliasing filter)
- Analog to digital converter (ADC)
- Digital computer or digital signal processor
- Digital to analog converter (DAC)
- Output lowpass filter (anti-imaging filter)



DSP System Components

- Analog input signal is filtered to be a band-limited signal by an input lowpass filter
- Signal is then sampled and quantized by an ADC
- Digital signal is processed by a digital circuit, often a computer or a digital signal processor
- Processed digital signal is then converted back to an analog signal by a DAC
- The resulting step waveform is converted to a smooth signal by a reconstruction filter called an anti-imaging filter

Advantages of DSP

- Versatility
  - Digital systems can be reprogrammed for other applications
  - Digital systems can be ported to different hardware
- Repeatability and stability
  - Digital systems can be easily duplicated
  - Digital systems do not depend on strict component tolerances
  - Digital system responses do not drift with temperature

Advantages of DSP (cont.)

- Simplicity
  - Some things can be done more easily digitally than with analog systems (e.g., linear phase filters)
  - Security can be introduced by encryption/scrambling
  - Digital signals easily stored on magnetic media without deterioration

Disadvantages of DSP

- DSP techniques are limited to signals with relatively low bandwidths
- The point at which DSP becomes too expensive will depend on the application and the current state of conversion and digital processing technology
  - Currently DSP systems are used for signals up to video bandwidths (about 10 MHz)
  - The cost of high-speed ADCs and DACs and the amount of digital circuitry required to implement very high-speed designs (> 100 MHz) makes them impractical for many applications
  - As conversion and digital technology improve, the bandwidths for which DSP is economical continue to increase

Disadvantages of DSP (cont.)

- The need for an ADC and DAC makes DSP not economical for simple applications (e.g., a simple filter)
- Higher power consumption and size of a DSP implementation can make it unsuitable for simple very low-power or small size applications

Applications of DSP

- **Speech Processing**
  - Noise filtering
  - Coding
    - 64 kbps-narrowband, 64 kbps-wideband
    - 32 kbps-narrowband, 32 kbps-wideband
    - 16 kbps-narrowband, 16 kbps-wideband
  - Compression
    - 64 kbps Mu-Law PCM
    - 32 kbps CCITT G.721 ADPCM
    - 16 kbps LD-CELP
    - 8 kbps CELP
    - 4.8 kbps CELP for STU-3
    - 2.4 kbps LPC-10E for STU-3
  - Recognition
  - Synthesis
  - Sampling rate changes

Applications of DSP

- **Image Processing**: enhancement, coding, compression, pattern recognition
- **Multimedia**: transmission of sound, still images, motion pictures, digital TV, video conferencing
- **Music**: recording, playback and manipulation (mixing, special effects), synthesis

Image Processing Example

Applications of DSP

- **Communication**: encoding and decoding of digital communication signals, detection, equalization, filtering, direction finding, echo cancellation
- **Radar and Sonar**: target detection, position and velocity estimation, tracking
- **Biomedical Engineering**: analysis of biomedical signals, diagnosis, patient monitoring, preventive health care, artificial organs

History of DSP

- Up to 1950’s: signal processing done with analog systems using electronic circuits or mechanical devices
- 1950’s: digital computers used to simulate signal processing systems before implementing in analog hardware – cheap way to vary parameters and test system output

History of DSP (cont.)

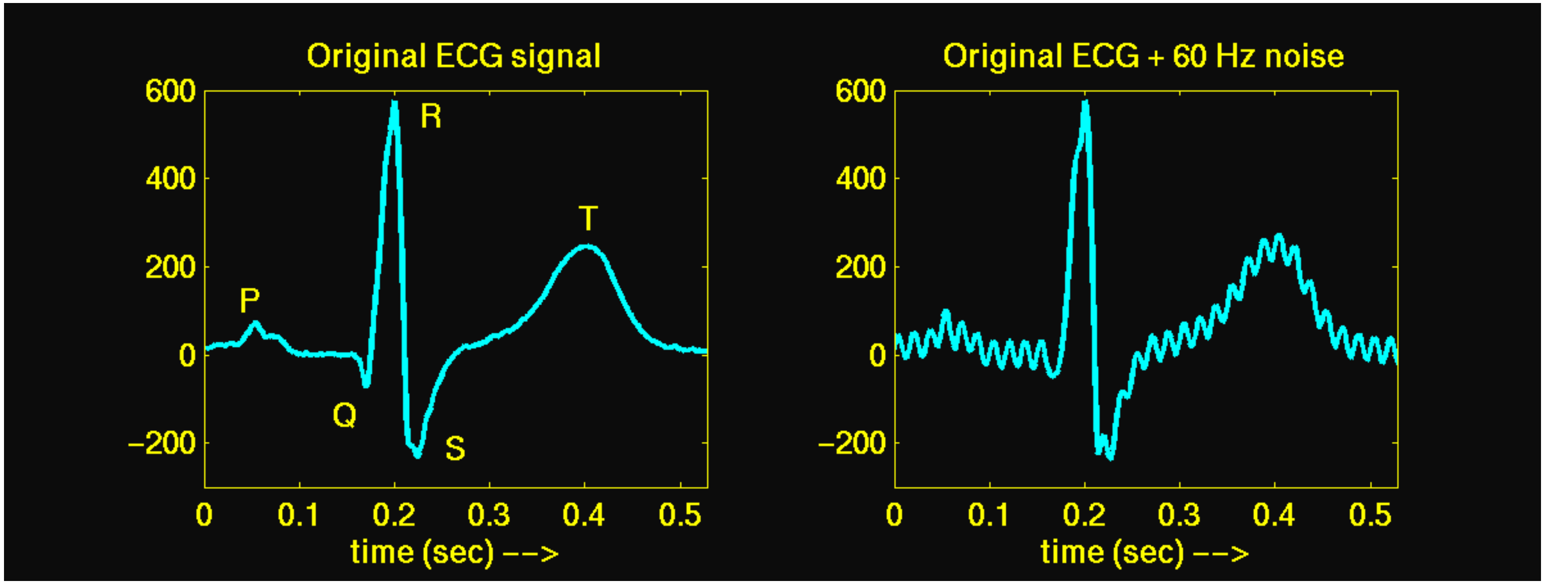
- 1965: Cooley and Tukey (re)discover efficient algorithm for Fast Fourier Transforms (FFTs) – made feasible real-time signal processing as well as algorithms previously thought impossible to implement on digital computers
- 1980’s: IC technology advancements led to very fast fixed-point and floating-point microprocessors for digital signal processing

DSP Functions

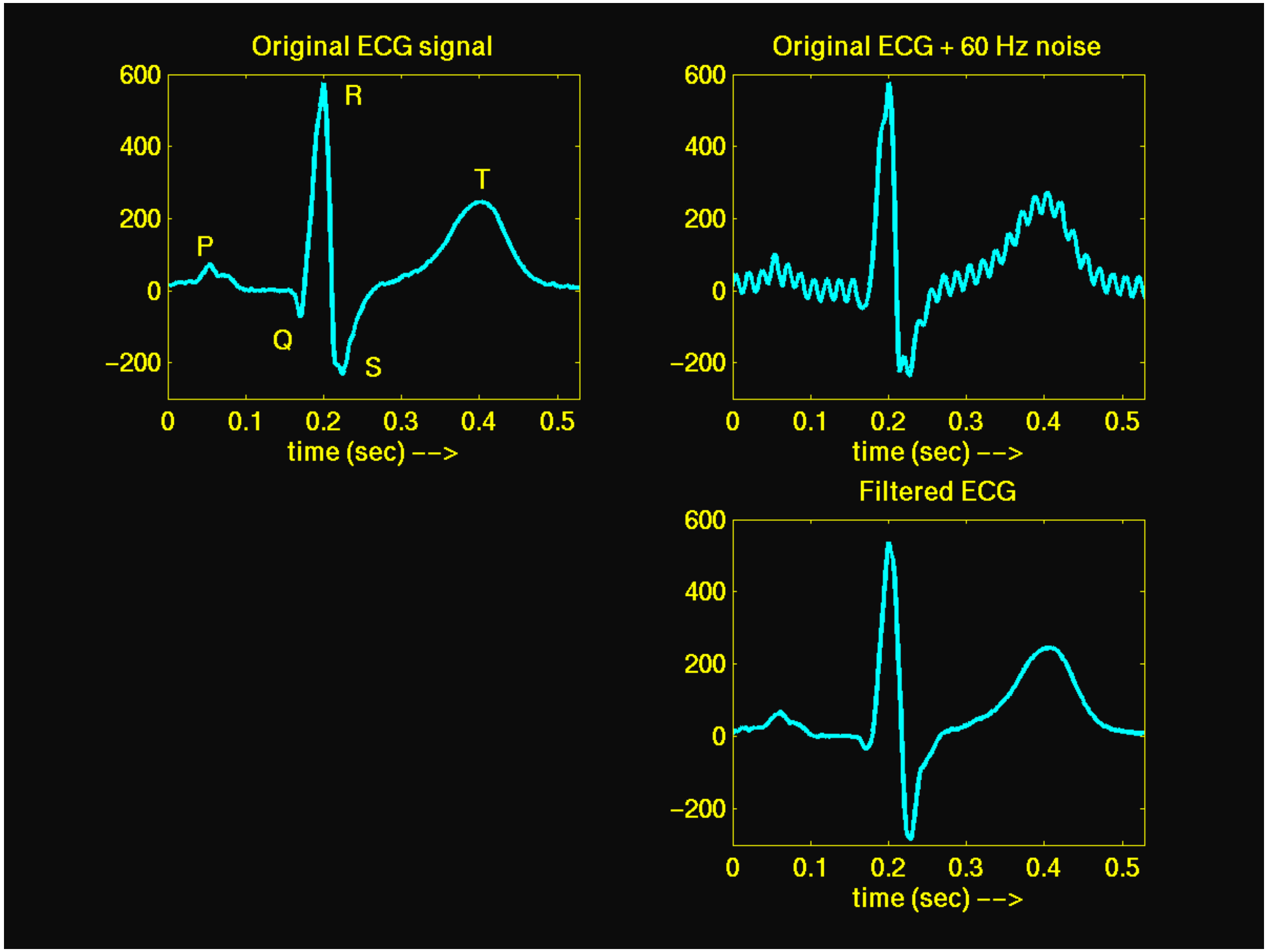
- Common features of DSP applications
  - They use a lot of multiplying and adding operations
  - They deal with signals that come from the real world
  - They require a certain response time
- Key DSP operations
  - Filtering
  - Correlation
  - Discrete transformation

Filtering Example

- Signals are usually a mix of “useful” information and noise
- How do we extract the useful information?
  - Filtering is one way



Filtering Example (cont.)

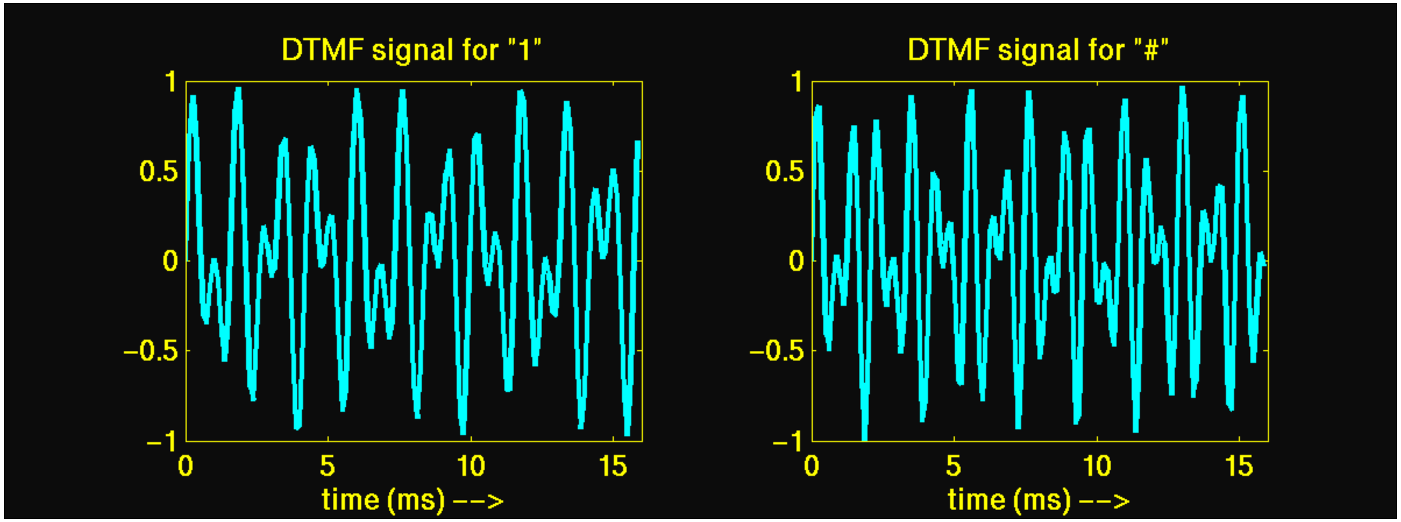


Filtering Equations

- Let $x[n]$ denote current input value (ECG+noise)
  - $x[n-1]$ is previous input value, $x[n-k]$ – k-th previous input
- Let $y[n]$ be the current filtered output value
  - $y[n-1]$ is previous output value , $y[n-k]$ – k-th previous output
- Filtering operations carried out for this example:

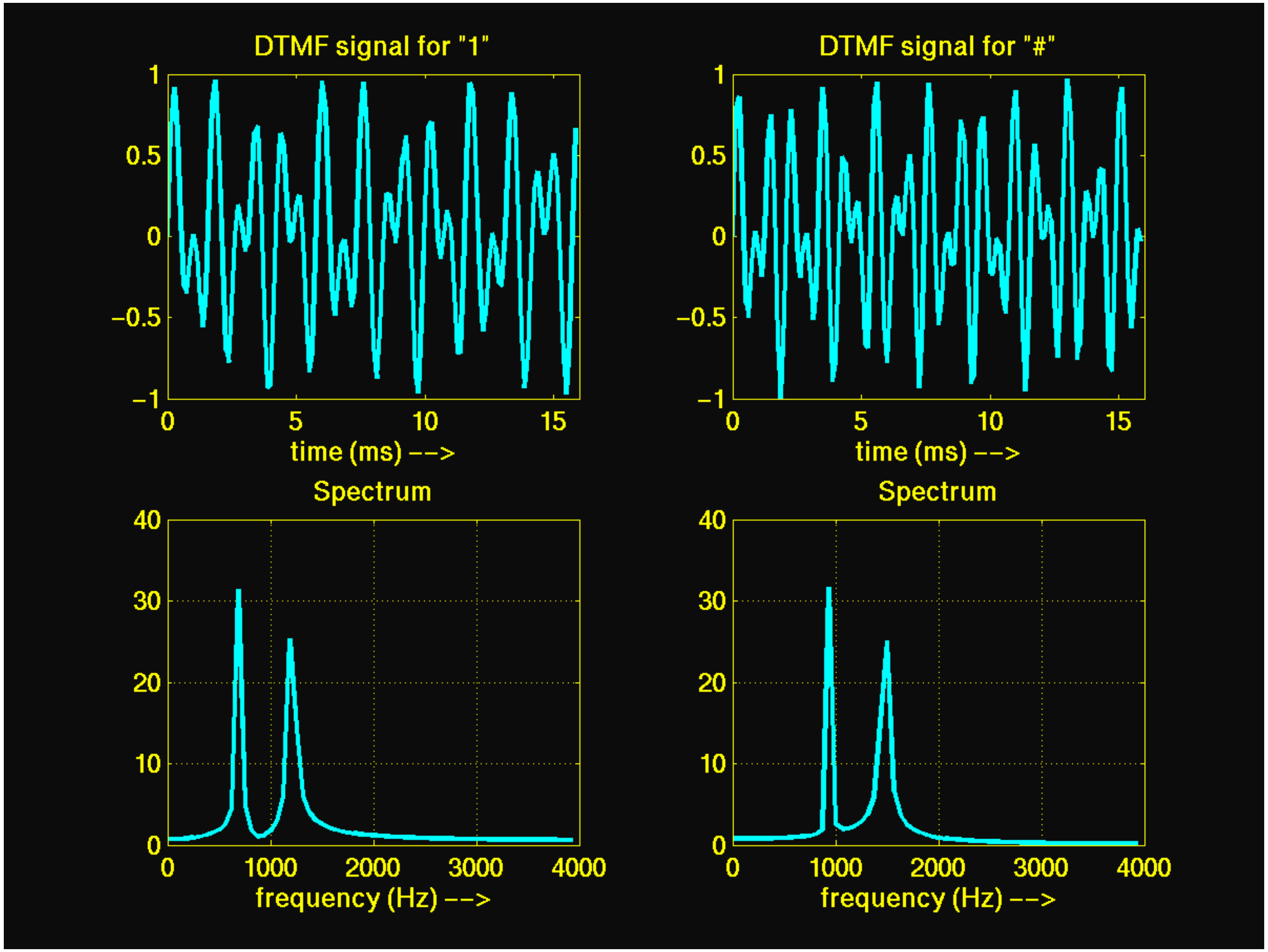
$$y[n] = 2.4*y[n-1] - 2.6*y[n-2] + 1.5\, y[n-3] - 0.4*y[n-4] + 0.6*x[n] - 1.9*x[n-1] + 2.8*x[n-2] - 1.9*x[n-3] + 0.6*x[n-4]$$

Transform Example



- Can you say which is “1” / ”#” by looking at them?
- If not, go to “frequency” domain
  - Another way to look at signals
  - Done using transforms

Transform Example (cont.)



Transform Equations

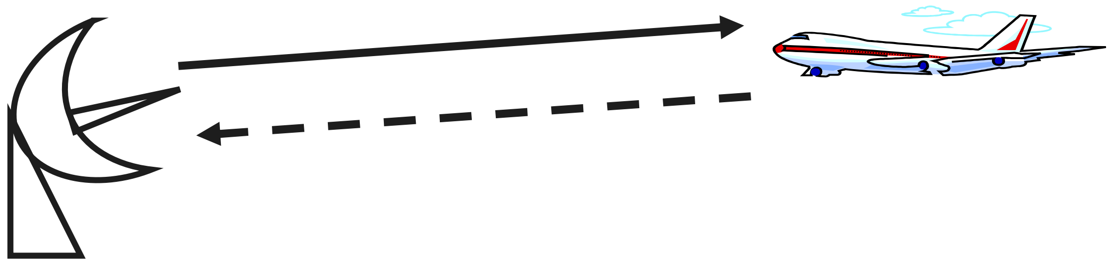
- Discrete Fourier Transform
- x – Time domain signal
- X – Frequency domain representation of x

$$X[k]=\sum_{n=0}^{N-1} x[n]e^{j(\frac{2\pi}{N})kn}, 0 \le k \le N-1$$

Correlation Example

- Provides a measure of similarity between 2 signals
- Typical application is locating a known signal
  - E.g., transmit a signal and see if you receive it back and also at what time you receive it back

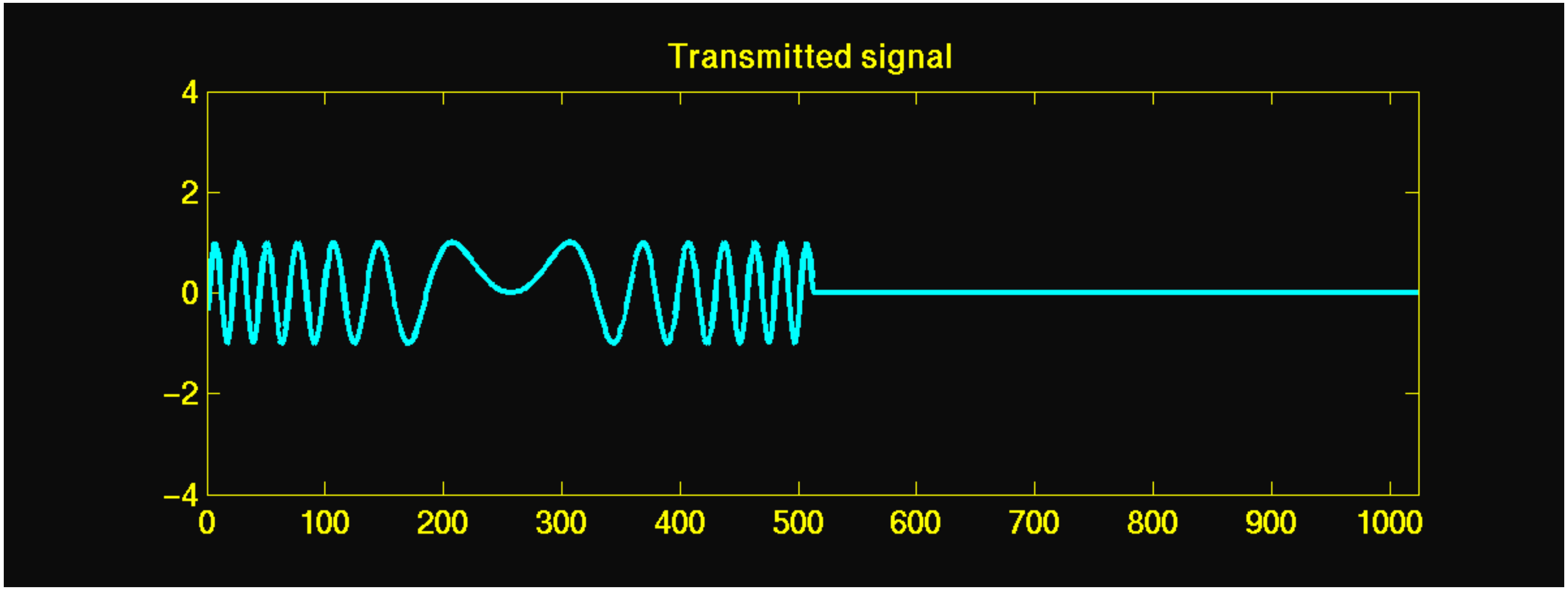
Radar



Blocked pipes!

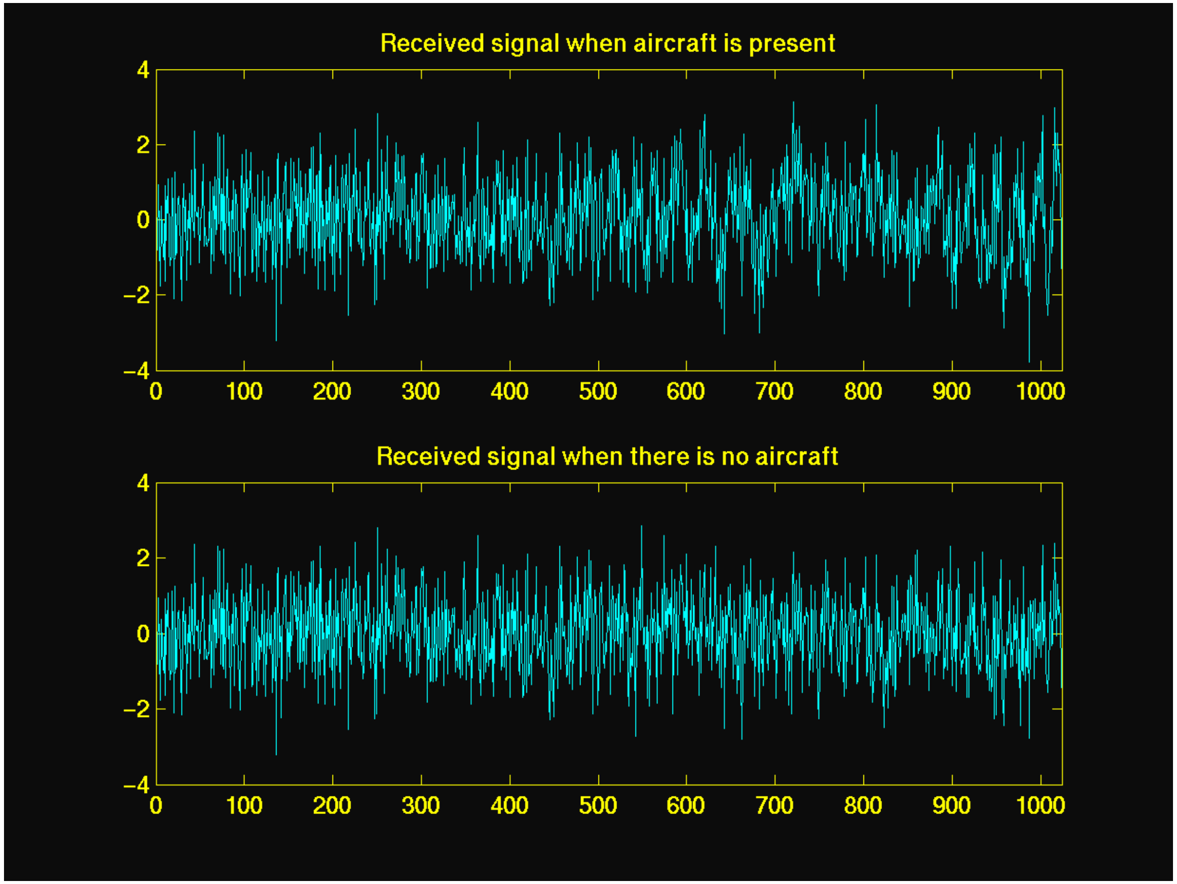
Correlation Example (cont.)

- Using radar, we transmit the signal shown below

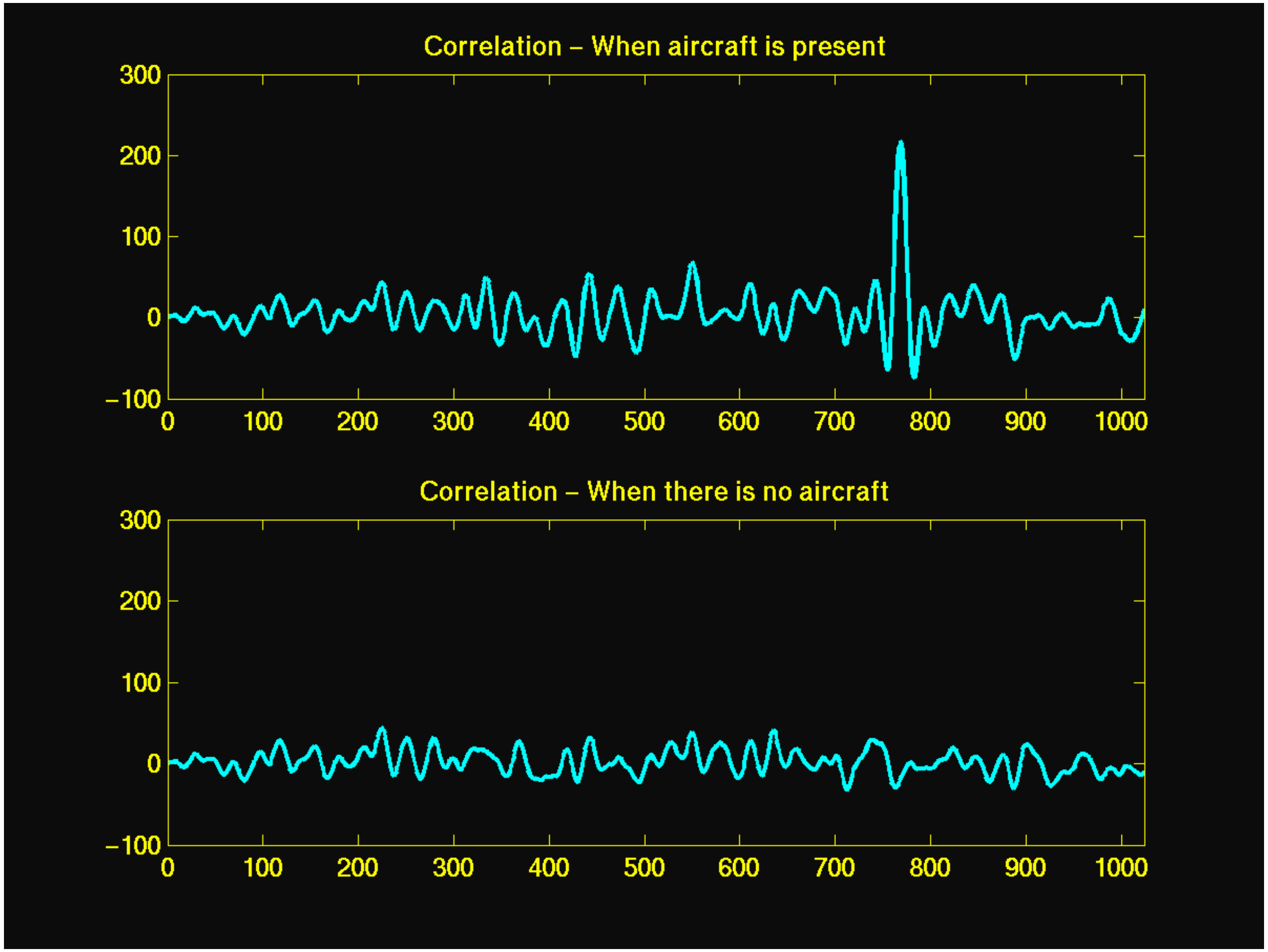


Correlation Example (cont.)

- We receive the following (note the noise!)



Correlation Example (cont.)



Correlation Equations

- Correlation
- x – Transmitted signal
- y – Received signal
- r_{xy} – Correlation coefficients

$$r_{xy}[l] = \sum_{n=-\infty}^{\infty} x[n]^* y[n-l], l = 0, \pm 1, \pm 2, \cdots$$

Why do we need DSPs?

- DSP operations require many calculations of the form: A = B*C + D
- This simple equation involves a multiply and an add operation
- The multiply instruction of a GPP is very slow compared with the add instruction
  - Motorola 68000 microprocessor uses
    - 10 clock cycles for add
    - 74 clock cycles for multiply

Why do we need DSPs? (cont.)

- Digital signal processors can perform the multiply and the add operation in just one clock cycle
  - Most DSPs have a specialized instruction that causes them to multiply, add and save the result in a single cycle
  - This instruction is called a MAC (Multiply, Add, and Accumulate)
- DSPs aim to minimize cost, power, memory use, and development time

Digital Signal Processor Architectures

- **Von Neuman**
  - Von Neuman machines store program and data in the same memory area with a single bus
  - An instruction contains the operation command and the address of data to be operated on (operand)
  - Most of the general-purpose microprocessors such as Motorola 68000 and Intel 80x86 use this architecture
  - It is simple in hardware implementation, but the data and program are required to share a single bus

Digital Signal Processor Architectures (cont.)

- **Harvard architecture**
  - The only difference in Harvard architecture is that program and data memories are separated and use physically separate transmission paths
  - Enables the machine to transfer instructions and data simultaneously-- enhances performance
  - The Harvard architecture is more commonly used in specialized microprocessors for real-time and embedded applications
  - However, only the early DSP chips use the Harvard architecture because of the cost

Digital Signal Processor Architectures (cont.)

- **Modified Harvard architecture**
  - Cost penalty with the Harvard architecture, which needs twice as many address and data pins on the chip
  - To balance cost and performance, modified Harvard architecture is used in most DSPs
  - Uses single data and address bus externally but internally there are two separate busses for program and data
  - The separation of program and data information is done by timing (multiplexing)
  - For one clock cycle, program information flows on the pins, and in second cycle data follows on the same pins

DSPs: Texas Instruments TMS320 Series

- **C1X, C2X**
  - Fixed-point devices with 16-bit data bus width
  - Used in toys, hard disk drives, modems and active car suspensions
- **C3X**
  - Floating-point devices with 32-bit data bus width, which provides much wider dynamic range as compared to fixed-point devices
  - Because of higher accuracy, used in hi-fi systems, voice mail systems and 3D graphic processing

DSPs: Texas Instruments TMS320 Series (cont.)

- **C4X**
  - 32-bit floating-point device designed for parallel processing
  - Optimized on-chip communication channel enables several devices to be put together to form a parallel cluster
  - Used in virtual reality, recognition and parallel processing systems
- **C5X**
  - Low power fixed-point DSPs
  - Used for personal and portable electronics such as cell phones, digital music players, and digital cameras

DSPs: Texas Instruments TMS320 Series (cont.)

- **C6X**
  - High performance DSPs, with speeds up to 1 GHz
  - Both fixed and floating-point devices
  - Used in wired and wireless broadband networks, imaging applications and professional audio
- **C8X**
  - Multimedia processors, with parallel processing on a single chip with advanced DSPs and a controlling RISC processor
  - Used in high performance telephony, 3D computer graphics, virtual reality and a number of multimedia applications

MATLAB

- MATLAB is an interactive, matrix-based system for scientific and engineering numeric computation and visualization
  - Strength: complex numerical problems can be solved easily with a programming language similar to C
  - Can be easily extended to create new commands and functions
  - Ideal software tool for studying digital signal processing
  - Graphing capability makes it possible to view results of processing and provide insight into complicated operations

MATLAB (cont.)

- Matlab programming is vector-based
  - Should *rarely* need to use loops
  - Can do most operations on vectors or matrices
  - E.g., in C:

```
for i = 1:10
  c(i) = a(i) + b(i)
  d(i) = a(i)*b(i)
end
```

in Matlab:

```
c = a+b
d = a.*b
```

MATLAB (cont.)

- Tips for Matlab programming of DSP
  - http://web.mit.edu/6.341/www/misc/matlab_primer_40.pdf
  - http://web.mit.edu/6.341/www/matlablinks.html
  - http://www.eedsp.gatech.edu/Information/MATLAB_User_Guide/index.shtml
  - http://www.eng.auburn.edu/~sjreeves/Classes/DSP/DSP.html

References for this Material

- Hong Kong City U Image Processing Lab's Introduction to DSP: www.ee.cityu.edu.hk/~lmpo/ee32211/notes/dsp/dsp.html
- BORES On-Line Introduction to DSP: www.bores.com/courses/intro/
- Dr. Budagavi's DSP Course: http://engr.smu.edu/~madhukar/ee5372.html
- Texas Intsruments: www.ti.com
- OGI ECE544: http://www.ece.ogi.edu/~macon/ECE544/
- Berkeley's EECS 20: http://robotics.eecs.berkeley.edu/~mayi/imgproc/