A1. Introduction to Signals

Objectives:

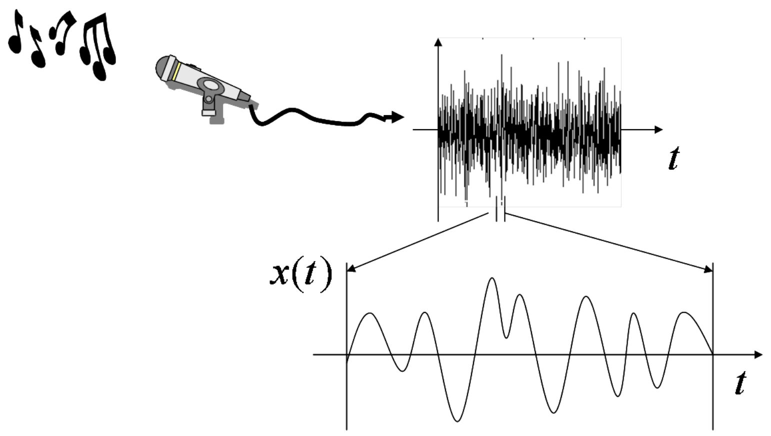
- Basic defintions of continuous time (analog) and discrete time (digital) signals
- Analog to Digital (Sampling) and Digital to Analog (Reconstruction) conversion
- Energy and Power of a Signal
- Signal to Noise Ratio

1. Basic Definitions

Note: you can either click on the "VIDEO: …" or cut and paste the full link in your browser.

VIDEO: Basic Definitions (23:29)
http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/section1-seg1_media/section1-seg1.wmv
http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/basicDefinitions.mp4

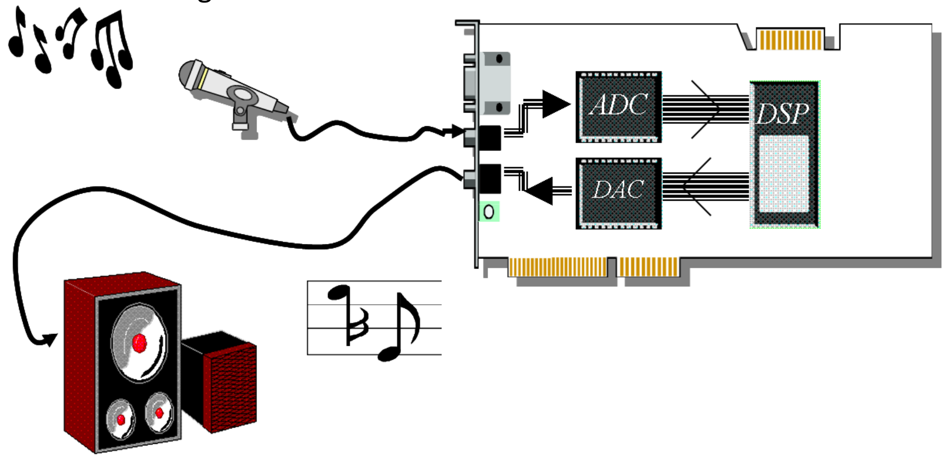
We all have an idea of what a signal is. We make an experiment, collect some data and we look at how a physical quantity (say the pressure of the air) evolves with time. We hear a sound and we can associate it to its waveform as we have seen in Disney's Fantasia.
A typical example is the signal from a microphone, as shown in the figure below.



Signal from a microphone.

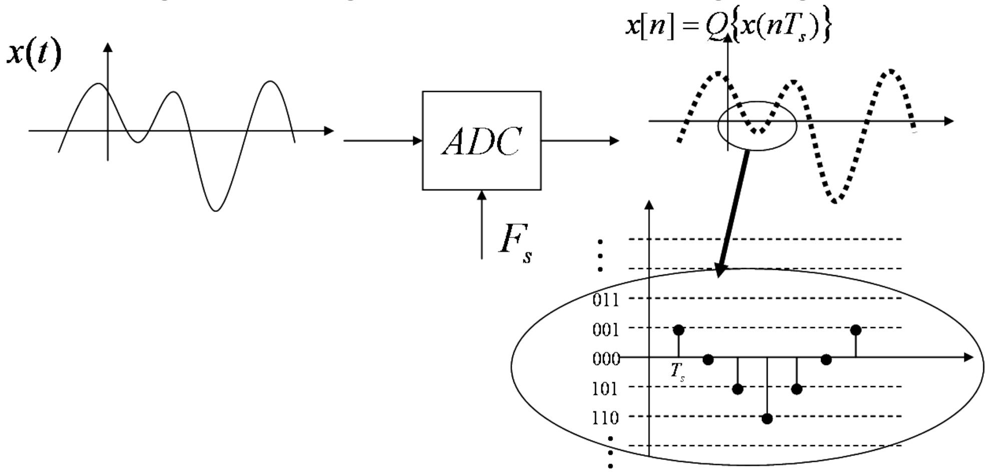
What we define as $x(t)$ is the value of the signal at time t. According to the physical meaning we attribute different units such as Volts, Amps, for electrical signals, or meter, meter per second and so on for mechanical signals.

In most applications we process signals numerically by a digital computer, a Digital Signal Processing (DSP) chip or any other digital device. In order to do so a signal has to be converted to a numerical sequence for processing. This is done by what is called an Analog to Digital Converter (ADC), which converts the signal from continuous time to discrete time. If we want to play back the result we use a Digital to Analog Converter (DAC) to convert a numerical sequence into a continuous time, physical signal. A general setup is shown in the figure below.



Digital Processing of a Signal

The Analog to Digital Converter (ADC) converts a continous time signal (also called Analog Signal) into a discrete time sequence by taking samples of the signal at discrete time instants. Since the numerical sequence is represented by a finite number of bits, the actual values of the numerical sequence are quantized into discrete amplitudes. For example with 8 bits per sample we have $2^8 = 256$ possible amplitudes and with 12 bits per sample we have $2^{12} = 4{,}096$ possibilities. Although in most applications there is sufficient number of bits to well rerpresent the data, there is always a small error due to this rounding effect. The figure below shows the Analog to Digital Converter.



Analog to Digital Converter: it samples a continuous time signal $x(t)$ ***into a discrete time sequence*** $x[n]$.

The Analog to Digital Converter (ADC) is characterized by the following parameters:

Sampling Frequency F_S **is the number of samples per second. It is expressed in** $Hz = 1/\sec$.

Sampling Interval $T_S = 1/F_S$ **is the time between samples. It is expressed in** $\sec$

Number of Bits per Sample N_B **is the number of bits representing every sample.**

Example. In a Compact Disk (CD) a signal is sampled at 44,100 samples per second. Every sample is represented by 16 bits. Therefore for every second of music we need

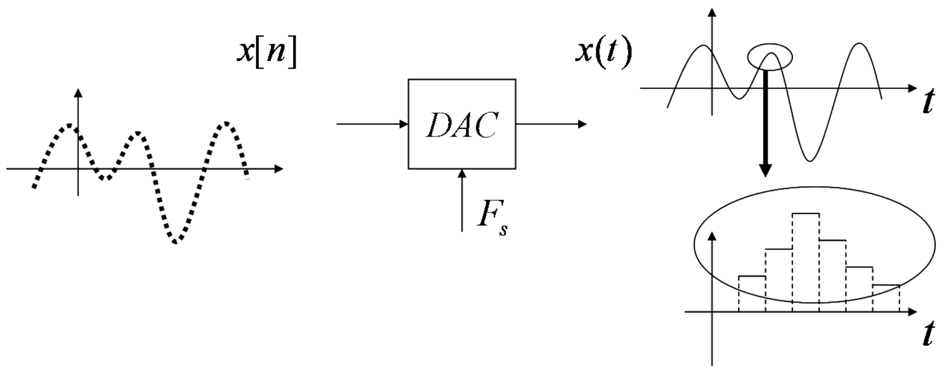
$$44{,}100 \times 16 = 705.6 kilobits/\sec = 88.2 kilobytes/\sec$$

Since a CD contains about 783 Megabytes of data, we can store

$$783 \times 10^6 / 88.2 \times 10^3 = 8{,}877 \sec = 148 \min$$

of a signal. Since most CD's are recorded in stereo (ie two signals for two speakers), a CD stores 148/2=74 minutes of music.

The Digital to Analog Converter (DAC) performs the reverse operation. It converts a numerical sequence $x[n]$ into a continuous time signal $x(t)$, so that we can drive (say) the speakers through an amplifier. The most common way of generating a continuous time signal out of a sampled numerical sequence is by "holding" the value constant within the sampling interval T_S, as shown in the figure below.



Digital to Analog Conversion: from a discrete time sequence $x[n]$ ***to a continuous time signal*** $x(t)$.

Apart from the continuous time and discrete time nature of the signals, notice the "physical" difference. The continuus time signal $x(t)$ is a physical signal, like a voltage, current, pressure … something with energy and power. You can drive speakers, move a vehicle and so on. Again it has "energy" in it. The discrete time signal $x[n]$ is a numerical sequence, just a bunch of bits into the computer, with no physical energy associated to it.

The physical concepts of Energy and Power of a signal are introduced next.

2. Energy and Power of a Signal

VIDEO: Energy of a Signal (01:33)

http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/section1-seg2_media/section1-seg2-0.wmv
http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/Energy_of_a_Signal.mp4

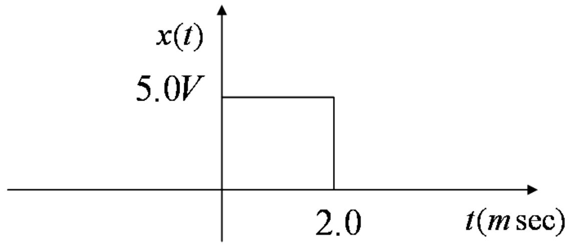
A continuous time signal $x(t)$ has energy associated to it. Mathematically we define it as follows:

Total Energy E_X **is the energy of a continuous time signal** $x(t)$ **defined as**

$$E_X = \int_{-\infty}^{+\infty} | x(t) |^2 \, dt$$

If the signal is (say) a voltage, than $x(t)$ is in $Volts$ and its energy in $Volts^2 \times \sec$.

Example. The signal shown below is a square pulse of 5.0 Volts of duration 2.0 milliseconds. So all the other values outside this interval of definition are zero. Its total energy is given by $E_X = (5.0)^2 \times 2.0 \times 10^{-3} = 50 \times 10^{-3} Volts^2 \times \sec$.



Signal for the Example

As we can see from the example, in order to have a finite energy a signal must converge to zero as time goes to infinity. This is fine for an isolated pulse, or a short burst. But if you have a signal which keeps going for a while (so that we can approximate it with an infinite length) the Energy becomes infinity, thus meaningless.

When a signal has a long (ideally infinite) duration, like a periodic pulse keeps repeating on and on (a bit like the Energizer Bunny which keeps going and going and going …) then clearly the energy is infinite, since it never decays to zero. In this case we define the following:

Average Power P_X **: Given a continuous time signal** $x(t)$ **its average power is defined as**

$$P_X = \lim_{T\to+\infty} \frac{1}{T} \int_{-T/2}^{+T/2} |x(t)|^2 \, dt$$

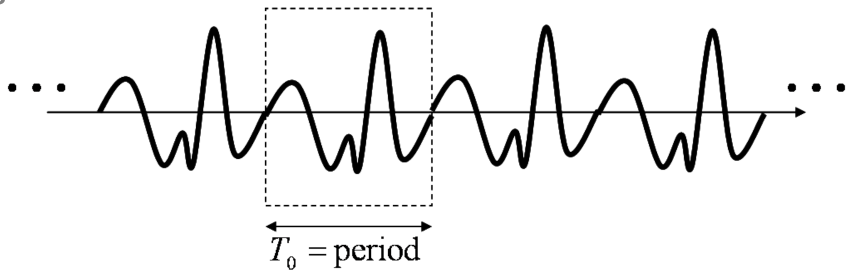
It is the average of the Total Energy over Time.

VIDEO: Power of a Signal (04:51)

http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/section1-seg2_media/section1-seg2-1.wmv
http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/Power_of_a_Signal.mp4

A particular case of interest is when the signal is a periodic repetition of a pulse, which repeats itself every T_0 seconds. This is called a periodic signal (more of this later) and an example is shown below.

Periodic Signal



A periodic signal with period T_0.

In this case the Power of the signal is the energy in one period divided by the length of the period. In other words

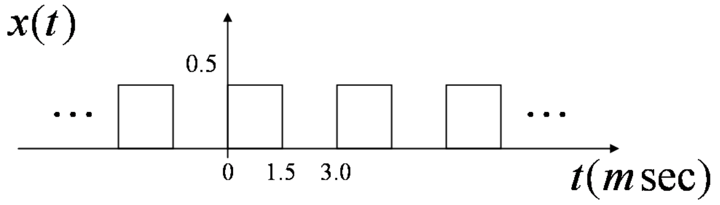
$$P_X = \frac{1}{T_0} \int_{\text{period}} |x(t)|^2 \, dt$$

where the integral is over one period.

Example. A continuous time signal $x(t)$ is a periodic repetition of a rectangular pulse as shown in the figure below. Clearly the total Energy is infinite, since it keeps repeating forever and never decays to zero. However the Power can be computed as

$$P_X = \frac{0.5^2 \times 1.5 \times 10^{-3}}{3 \times 10^{-3}} = 0.125 \text{ Volts}^2$$

Example of Periodic Signal



Periodic Rectangular Pulse of the Example.

Root Mean Square (RMS) of a Signal is the square root of the signal power, ie

$$X_{RMS} = \sqrt{P_X}$$

VIDEO: deciBells (07:51)

http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/section1-seg2_media/section1-seg2-2.wmv
http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/deciBells.mp4

In most applications what is relevant is not the absolute power of a signal, but a relative power, with respect to a reference signal. In particular, when this the case, we express it on a logarithmic scale called "decibel". In particular we define the following:

Relative Power in decibels (dB's).

Given a signal $x(t)$ with power P_X, and a reference signal $x_{ref}(t)$ with power P_{Xref}, the relative power is defined as

$$P_X\big|_{dB} = 10\log_{10}\frac{P_X}{P_{Xref}}$$

Example: back to the previous example, let a reference signal have power $P_{Xref} = 0.01\,Volts^2$. The relative power in dB's is computed as

$$P_X\big|_{dB} = 10\log_{10}\frac{P_X}{P_{Xref}} = 10\log_{10}\frac{0.125}{0.01} = 10.97dB\,.$$

Specially for acoustic signals, the values in dB's are very important and sort of a standard. Let's see in this case how these values are defined. The strength of a sound depends on the power transmitted through the air per unit surface. This will excite the membrane of our ears so that we perceive the sound. Although each person is different a standard value for the threshold of perception is

$$P_{threshold} = 10^{-12}\,Watts/m^2$$

Any sound with energy lower than the threshold most likely will not be perceived by most people. It is customary to use this as the reference power. The Table below shows the power and dB's for a number of signal intensities, using a music notation. In this case as ppp (pianissimo, very low and faint), p (piano, low), f (forte, loud), fff (fortissimo, very loud) and pain (lots of pain!).

	$Watts/m^2$	dB
Threshold	10^{-12}	0
ppp	10^{-8}	40
p	10^{-6}	60
f	10^{-4}	80
fff	10^{-2}	100
pain	1	120

3. Signal to Noise Ratio

VIDEO: Signal to Noise Ratio (07:53)

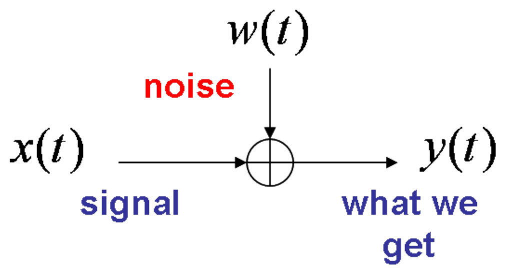
http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/section1-seg2_media/section1-seg2-3.wmv
http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/signalNoise.mp4

In most applications of signal processing we make a distinction between the signal we want (ie which carries useful information) from the signal we don't want, also called the disturbance. For example, we talk to a person and we hear the person him(her)self and the background noise. The latter comes from a numer of sources, typically cars passing by, other people in the crowd, costruction and so on. In this case the "desired" signal is what we hear from our friend (even if what (s)he says might not be so desirable) and the noise is everything else which precludes a good perception.

The goal is to measure how much noise there is. Of course the "how much" is relative, since if we talk to a person who wispers, just a little bit of noise is disturbing. But if we talk to someone who has a loud voice, we can tolerate a Boeing 747 landing next to us and still understand it!

A diagram of this is shown below.

Signal plus Noise



What we receive in general is the Signal which carroes information, plus all other disturbances which we group into Noise.

In this situation we define the following:

Signal to Noise Ratio (SNR). Given a signal $x(t)$ with power P_X and noise $w(t)$ with power P_W we define the Signal to Noise Ratio (in dB's) as

$$SNR = 10\log_{10}\frac{P_X}{P_W} = P_X\big|_{dB} - P_W\big|_{dB}$$

Let's refer to the table of acoustic signals above and see some examples. We all have an idea what piano, forte and so on sound like. Just think of the volume control of your stereo, when you keep low not to wake up the baby (piano), or increase the volume to entertain your friends (forte) or to the point that your neighbor calls the police (fortissimo).

Example You are at a concert and during a piece "forte" someone next to you talks "pianissimo", on a whisper. You might hear the voice a little bit but still you can hear the orchestra very well. Most people do not get annoyed since you can bearly hear it.
In this case the Signal top Noise Ratio is $SNR = P_{forte} - P_{pianissimo} = 80 - 40 = 40dB$.

Example At the same concert and during another piece "piano" the same person keeps whispering "pianissimo". Now you try to pay attention to the music, which you can still hear. But now the Signal to Noise Ratio is lower, down to
$SNR = P_{piano} - P_{pianissimo} = 60 - 40 = 20dB$.

Example Now the music goes "pianissimo" and the same person keeps whispering "pianissimo". Now you are ennoyed since the two signals (music and noise) have comparable power (ie loudness) and the Signal to Noise Ratio is
$SNR = P_{pianissimo} - P_{pianissimo} = 40 - 40 = 0dB$.

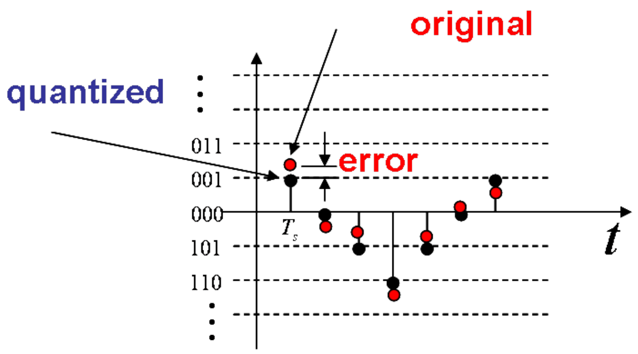
From the examples above you get an idea of a high SNR (40dB) where the signal is very clear, a medium SNR (20dB) where the signal is clear and you hear the disturbance a little bit, to a case where signal and noise are comparable (SNR=0dB or lower) which really affects the perception of the signal. If the power of the noise is even more than the power of the signal, then the SNR becomes negative.

4. SNR and Quantization Noise

VIDEO: Quantization Noise (9 :12)

http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/section1-seg2_media/section1-seg2-4.wmv
http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/quantizationNoise.mp4

Recall that when we convert a signal from continuous time to discrete time (ie we digitize it) we sample it in time into a numerical sequence and also we quantize the amplitude. This is due to the fact that computers and digital machines only understand numbers (that's why we take samples). Also every number has a finite precision and it is represented only by a finite number of bits. In other words the amplitude of a signal at each sample gets "rounded" to the closest level. This is shown in the figure below.



Sampled original signal vs sampled and quantized signal.

The quantization process is defined by the number of Bits per Sample N_B.

Example. If we choose $N_B = 8$, that is to say 8 bits per sample, then the total number of quantization levels is $2^8 = 256$. The values can be all positive as 0,…,255 or negative and positive as -128, … ,-1,0,+1,…,+127. In the first case we say that the values are unsigned while in the second case they are signed.

Example. By choosing $N_B = 16$ bits per sample, the total number of quantization levels is much higher and it is given by $2^{16} = 65{,}536$ which again can be all positive unsigned (0,…,65,535) or signed (-32,768,…,-1,0,+1,…,+32,667)..

Looking at these two examples, clearly with 16 bits per sample we have a lot more possibilities and the quantized signal is closer to the original signal. The quantization error can be considered as noise, since it is just a disturbance added to the signal. Therefore we want to characterize it as a Signal to Noise Ratio, so we get an idea of how much of a disturbance it is.

At this point we need a bit of leap of faith, since the arguments are statistical and beyond the scope of this course. However a numerical idea yields some useful information on how we decide on the number of bits per sample.

Statistically we can assume that the error due to quantization is uniformly distributed within the quantization interval. This leads to the following fact.

Consider a signal $x(t)$ with power P_X sampled and quantized with N_B bits per sample. Then if the signal is equally distributed over all the quantization levels, the power of the quantization level can be estimated as

$$P_{error} = \frac{1}{3 \times 2^{2N_B}} P_{signal}$$

As a consequence, the Signal to Noise Ratio (SNR) due to the quantization is given by

$$SNR = 10\log_{10}\left(3 \times 2^{2N_B}\right) = 4.77 + 6.02 N_B \ dB$$

This expresses the quantization error in terms of Signal to Noise Ratio, when we use N_B bits per sample. It is particularly usefull when we design a Digital Signal Processing system, in order to estimate the number of bits per sample.

Example. Suppose we want to have a very clean signal, so that the quantization noise has a SNR of at least 100dB. Using the formula above, we can determine the number of bits per sample N_B from the expression $4.77 + 6.02 N_B \geq 100$. This yields $N_B \geq 95.23/6.02 = 15.8$. Since it has to be an integer, we need at least 16 bits per sample.

5. Introduction to Matlab

This is an introduction to Matlab in order to gain basic knowledge, applied to this course. For ease of explanation, it is shown as two recorded hands on lectures.

5.1 Matlab Basics

The goal of this lecture is how to get started with Matlab. In particular we begin with Matlab as a calculator for real and complex arithmetics. Then we introduce arrays (matrices and vectors) as the most important data structures in Matlab. Basic element by element operations between arrays are introduced, together with how to plot the results and “sound” them on the audio card of the computer.

Topics:

- Matlab Workspace (02:11)

http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-0.wmv
http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-0.mp4

- Basic Arithmetics (00:59)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-1.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-1.mp4

- `whos` (02:55)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-2.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-2.mp4

- Complex Numbers (08:27)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-3.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-3.mp4

- Matrices and Vectors (10:38)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-4.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-4.mp4

- Elementary Matrix Operations (06:30)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-5.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-5.mp4

- `plot` and `stem` (04:41)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-6.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-6.mp4

- `sound` (02:18)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-7.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-7.mp4

- `transpose` (12:39)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-8.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-basics-8.mp4

5.2 Matlab Applications

In this lecture we show how to import an audio file into Matlab, so it can be manipulated and processed. In particular we see how to add noise to a signal to simulate a given Signal to Noise Ratio (SNR) and play the signal in the sound card.

Topics:

- WAV files (05:00)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-applications-0.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-applications-0.mp4

- wavread (04:20)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-applications-1.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-applications-1.mp4

- time indexing (09:50)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-applications-2.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-applications-2.mp4

- Power and SNR (26:23)
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-applications-3.wmv
  http://faculty.nps.edu/rcristi/eo3404/a-signals/videos/matlab-applications-3.mp4