Information and Signal Theory

Collection Editor:
Anders Gjendemsjø

Information and Signal Theory

Collection Editor:

Anders Gjendemsjø

Authors:

Behnaam Aazhang
Richard Baraniuk
Catherine Elder
Anders Gjendemsjø
Stephen Kruzick
Ricardo Radaelli-Sanchez
Melissa Selik

Online:

< http://cnx.org/content/col10211/1.19/ >

C O N N E X I O N S

Rice University, Houston, Texas

Collection structure revised: August 3, 2006
PDF generated: September 29, 2011
For copyright and attribution information for the modules contained in this collection, see p. 89.

Table of Contents

Cover Page[1]

TTT4110: Information & Signal theory

NOTE: Analysis and processing of signals that carry information. Representation of signals in time and frequency domain.

Instructor: Bojana Gajic[2] **Scientific Assistants**: Sebastien de la Ketthulle[3] , Anders Gjendemsjø[4]
Course Webpage: NTNU TTT4110[5]

Welcome to **TTT4110: Information & Signal Theory** Connexions[6] pages. At these pages we will present the following topics:

- Signals (Section 1.1)
- Convolution (Section 2.1)
- the Sampling Theorem (Section 4.1)
- Basic Information Theory (Section 5.1)
- Filters (Chapter 7)
- Decibel with DSP applications (Chapter 6)

The material in these pages are partly based on the book **Representing Information by Signals,4th edition,** by Tor Ramstad.

[1]This content is available online at <http://cnx.org/content/m11441/1.19/>.
[2]http://www.tele.ntnu.no/users/bojana/
[3]delaketh@iet.ntnu.no__NOSPAM
[4]gjendems@iet.ntnu.no__NOSPAM
[5]http://www.tele.ntnu.no/signal/courses/infosig
[6]http://cnx.rice.edu

Chapter 1

Basic properties of signals

1.1 Introduction[1]

To describe signals and to understand that signals can carry information we need tools for mathematical description and manipulation of signals.

In this chapter we introduce several important signals and show simple methods of describing them. Depending on which type of signals we are looking at, it will be different methods availiable for manipulating them. The elementary operations for manipulating signals and sequences will be described.

Contents of this chapter

- Introduction (current module)
- Discrete time signals (Section 1.2)
- Analog signals (Section 1.3)
- Discrete vs Analog signals (Section 1.4)
- Frequency definitions and periodicity (Section 1.5)
- Energy & Power (Section 1.6)
- Exercises (Section 1.7)

The simplest signals are one-dimensional and what follows is a classification of them.

1.1.1 Classification of signals

1.1.1.1 Analog signals

An **analog signal** is a continuous function of a continuous variable. Referring to Figure 1.1, this corresponds to that both the 1st AND the 2nd axis is continuous. The 1st axis will in general correspond to the variable t, meaning time. In this context we define

- signal range - the possible amplitude values the signal can take
- signal axis - the time interval for which the signal exists

[1]This content is available online at <http://cnx.org/content/m11479/1.9/>.

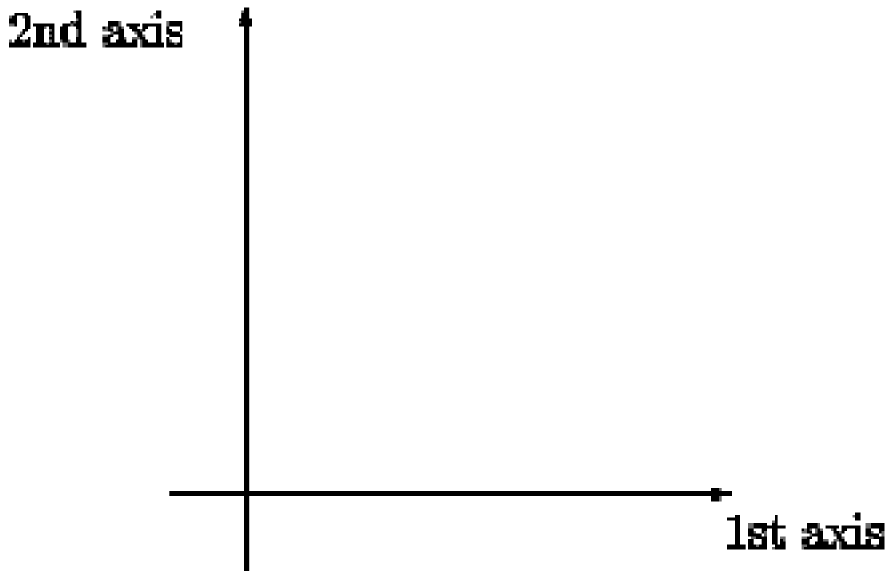


Figure 1.1: Reference axes

1.1.1.2 Time discrete signals

A **time discrete signal** is a continuous signal of a discrete variable. Referring to Figure 1.1, we have the 1st axis discrete while the 2nd axis is continuous. Often we assign the values of the 1st axis to a variable n. Time discrete signals often originate from analog signals being sampled. More on that in the Sampling theorem (Section 4.1) chapter.

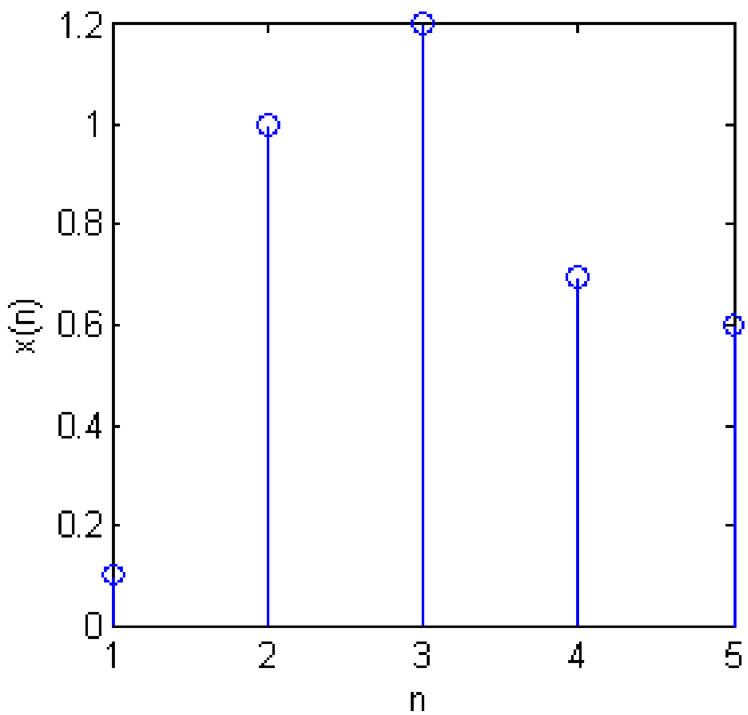


Figure 1.2: Time discrete signal

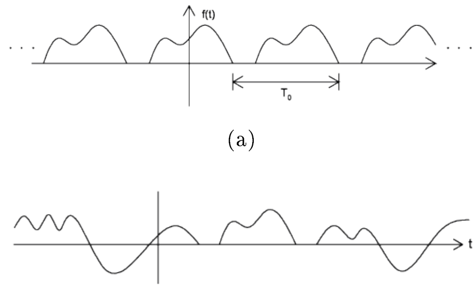
Note that the signal is only defined for integer values along the 1st axis. We do not have any information other than the values at index points.

1.1.1.3 Digital signals

Let the signal be a discrete function of a discrete variable, e.g. 1st and 2nd axis discrete, then the signal will be **digital**. Examples of digital signals are a binary sequence. Digital signals often arise from sampling analog signals and the samples being assigned to a discrete value.

1.1.1.4 Periodic vs non periodic signals

All the signals mentioned above can be periodic. For time discrete and digital signals one has to be extra cautious when "declaring" periodicity as we will see in Frequency definitions & periodicity (Section 1.5). Figure 1.3 shows a periodic signal with period T_0 and an aperiodic signal.



(a)

(b)

Figure 1.3: (Figures by Melissa Selik) (a) Periodic signal (b) Aperiodic signal

1.1.2 Matlab file

time_discrete.m[2]

1.1.3

Take a look at Discrete time signals (Section 1.2); Analog signals (Section 1.3); Discrete vs Analog signals (Section 1.4); Frequency definitions and periodicity (Section 1.5); Energy & Power (Section 1.6); Exercises (Section 1.7) ?

1.2 Discrete time signals[3]

The signals and relations presented in this module are quite similar to those in the Analog signals (Section 1.3) module. So do compare and find similarities and differences!

1.2.1 Sequences

Generally a time discrete signal is a **sequence** of real or complex numbers. Each component in the sequence is identified by an index: ...x(n-1),x(n), x(n+1),...

Example 1.1
[x(n)] = [0.5 2.4 3.2 4.5] is a sequence. Using the index to identify a component we have $x(0) = 0.5$, $x(1) = 2.4$ and so on.

1.2.2 Manipulating sequences

Addition - Add individually each component with similar index
Multiplication by a constant - Multiply every component by the constant
Multiplication of sequences - Multiply each component individually
Delay - A delay by k implies that we shift the sequence by k. For this to make sense the sequence has to be of infinite length.

[2]http://cnx.rice.edu/content/m11479/latest/time_discrete.m
[3]This content is available online at <http://cnx.org/content/m11476/1.16/>.

Example 1.2
Given the sequences [x(n)] = [0.5 2.4 3.2 4.5] and [y(n)] = [0.0 2.2 7.2 5.5].
a)Addition. [z(n)]=[x(n)]+[y(n)]=[0.5 4.6 10.4 10.0]
b)Multiplication by a constant c=2. [w(n)]= 2 *[x(n)] = [1.0 4.8 6.4 9.0]

1.2.3 Elementary signals & relations

1.2.3.1 The unit sample

The **unit sample** is a signal which is zero everywhere except when its argument is zero, then it is equal to 1. Mathematically

NOTE: $\delta(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases}$

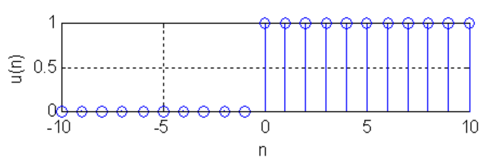
The unit sample function is very useful in that it can be seen as the elementary constituent in any discrete signal. Let $x(n)$ be a sequence. Then we can express $x(n)$ as follows (using the unit sample definition and the delay operation)

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k) \tag{1.1}$$

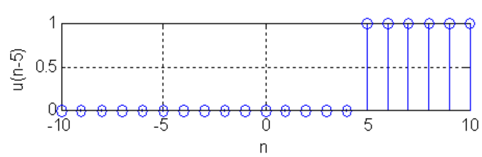
1.2.3.2 The unit step

The **unit step** function is equal to zero when its index is negative and equal to one for non-negative indexes, see Figure 1.4 for plots.

NOTE: $u(n) = \begin{cases} 1 & \text{if } n \geq 0 \\ 0 & \text{otherwise} \end{cases}$



(a)



(b)

Figure 1.4: Two unit step functions. (a) Unit step function, no delay. (b) Unit step function, delayed by 5.

1.2.3.3 Trigonometric functions

The discrete **trigonometric** functions are defined as follows. n is the sequence index and ω is the angular frequency. $\omega = 2\pi f$, where f is the digital frequency.

NOTE: $x(n) = \sin(\omega n)$

NOTE: $x(n) = \cos(\omega n)$

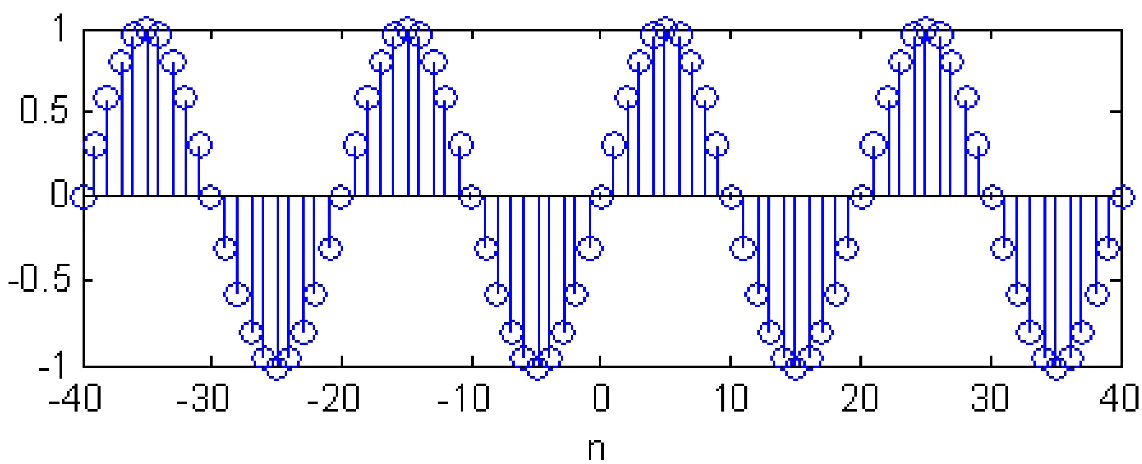


Figure 1.5: A discrete sine with digital frequency 1/20.

1.2.3.4 The complex exponential function

The **complex exponential** function is central to signal processing and some call it **the** most important signal. Remember that it is a sequence and that $j = \sqrt{-1}$ is the imaginary unit.

NOTE: $x(n) = e^{j\omega n}$

1.2.4 Euler's relations

The **complex exponential** function can be written as a sum of its real and imaginary part.

$$x(n) = e^{j\omega n} = \cos(\omega n) + j\sin(\omega n) \tag{1.2}$$

By complex conjugating (1.2) and add / subtract the result with (1.2) we obtain Euler's relations.

NOTE: $\cos(\omega n) = \frac{e^{j\omega n} + e^{-(j\omega n)}}{2}$

NOTE: $\sin(\omega n) = \frac{e^{j\omega n} - e^{-(j\omega n)}}{2j}$

The importance of Euler's relations can hardly be stressed enough.

1.2.5 Matlab files

unit_step_discrete.m[4]

1.2.6

Take a look at Introduction (Section 1.1); Analog signals (Section 1.3); Discrete vs Analog signals (Section 1.4); Frequency definitions and periodicity (Section 1.5); Energy & Power (Section 1.6); Exercises (Section 1.7) ?

1.3 Analog signals[5]

The signals signals and relations presented in this module are quite similar to those in the Discrete time signals (Section 1.2) module. So do compare and find similarities and differences!

1.3.1 Manipulating signals

Mathematical operations on analog signals are unambiguous. We require that the signals are defined over the same time interval when using operations such as addition, multiplication, division and so on.

1.3.2 Elementary signals & relations

1.3.2.1 The (Dirac) delta function

The **delta function** is a peculiar function that has zero duration, infinite height, but still unit area! Mathematically we have the following two properties

NOTE: $\delta(t) = 0$ for $t \neq 0$

NOTE: $\int_{-\infty}^{\infty} \delta(t)\, dt = 1$

The delta function has a useful property, namely the **sampling property**.

$$x(t) = \int_{-\infty}^{\infty} x(\tau)\, \delta(t-\tau)\, d\tau \qquad (1.3)$$

At this stage this may seem not particulary useful, so for now just convince yourself that the above relation holds.

(We assume that $x(t)$ is "well behaved" at $t = \tau$, that is continuous and finite.)

1.3.2.2 The unit step function

The **unit step** function is equal to zero when its argument is negative and equal to one for non-negative arguments, see Figure 1.6 for plots.

NOTE: $u(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$

[4]http://cnx.rice.edu/content/m11476/latest/unit_step_discrete.m

[5]This content is available online at <http://cnx.org/content/m11478/1.8/>.

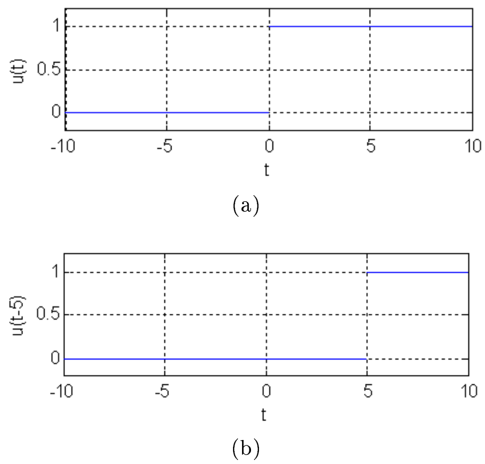


Figure 1.6: Two unit step functions. (a) Unit step function, no delay. (b) Unit step function, delayed by 5.

1.3.2.3 Trigonometric functions

The **trigonometric** functions are central to signal processing and telecommunications. They are defined as follows, where Ω is the angular frequency. $\Omega = 2\pi F_0$, where F_0 is the **frequency** of the signal.

NOTE: $x(t) = \sin(\Omega t)$

NOTE: $x(t) = \cos(\Omega t)$

See also Frequency definitions & periodicity (Section 1.5).

1.3.2.4 The complex exponential function

The **complex exponential** function is central to signal processing and some call it **the** most important signal. $j = \sqrt{-1}$ is the imaginary unit.

NOTE: $x(t) = e^{j\Omega t}$

1.3.3 Euler's relations

The **complex exponential** function can be written as a sum of its real and imaginary part.

$$x(t) = e^{j\Omega t} = \cos(\Omega t) + j\sin(\Omega t) \tag{1.4}$$

By complex conjugating (1.4) and add / subtract the result with (1.4) we obtain Euler's relations.

NOTE: $\cos(\Omega t) = \frac{e^{j\Omega t} + e^{-(j\Omega t)}}{2}$

NOTE: $\sin(\Omega t) = \frac{e^{j\Omega t} - e^{-(j\Omega t)}}{2j}$

The importance of Euler's relations can hardly be stressed enough.

1.3.4 Matlab file

unit_step_analog.m[6]

1.3.5

Take a look at Introduction (Section 1.1); Discrete time signals (Section 1.2); Discrete vs Analog signals (Section 1.4); Frequency definitions and periodicity (Section 1.5); Energy & Power (Section 1.6); Exercises[7] ?

1.4 Discrete vs Analog[8]

When comparing analog vs discrete time, we find that there are many similarities. Often we only need to substitute the varible t with n and integration with summation. Still there are some important differences that we need to know. As the complex exponential signal is truly central to signal processing we will study that in more detail.

1.4.1 Analog

The complex exponential function is defined: $x(t) = e^{j\Omega t}$. If Ω(rad/second) is increased the rate of oscillation will increase continuously. The complex exponential function is also periodic for **any** value of Ω. In figure Figure 1.7 we have plotted $e^{j\pi t}$ and $e^{j3\pi t}$ (the real parts only). In Figure 1.7 we see that the red plot, corresponding to a higher value of Ω, has a higher rate of oscillation.

[6] http://cnx.rice.edu/content/m11478/latest/unit_step_analog.m
[7] "Existence of the Minimum Variance Unbiased Estimator (MVUB)" <http://cnx.org/content/m11428/latest/>
[8] This content is available online at <http://cnx.org/content/m11527/1.10/>.

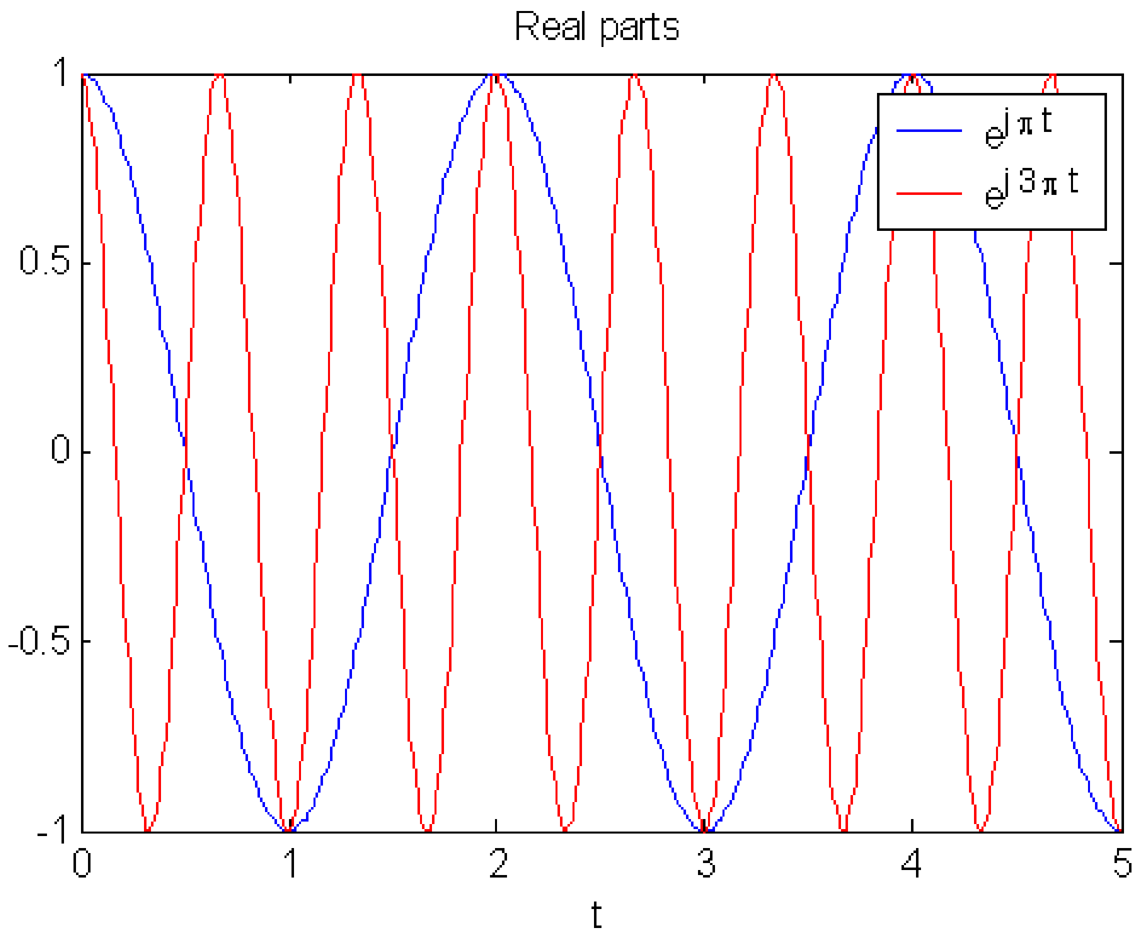


Figure 1.7: Real parts of complex exponentials.

1.4.2 Discrete time

The discrete time complex exponential function is defined: $x(n) = e^{j\omega n}$.

If we increase ω (rad/sample) the rate of oscillation will increase and decrease periodically. The reason is: $e^{j(\omega+2\pi k)n} = e^{j\omega n}e^{j2\pi kn} = e^{j\omega n}$, where $n, k \in \mathbb{Z}$.

This implies that the complex exponential with digital angular frequency ω is identical to a complex exponential with $\omega_1 = \omega + 2\pi$, see Figure 1.8

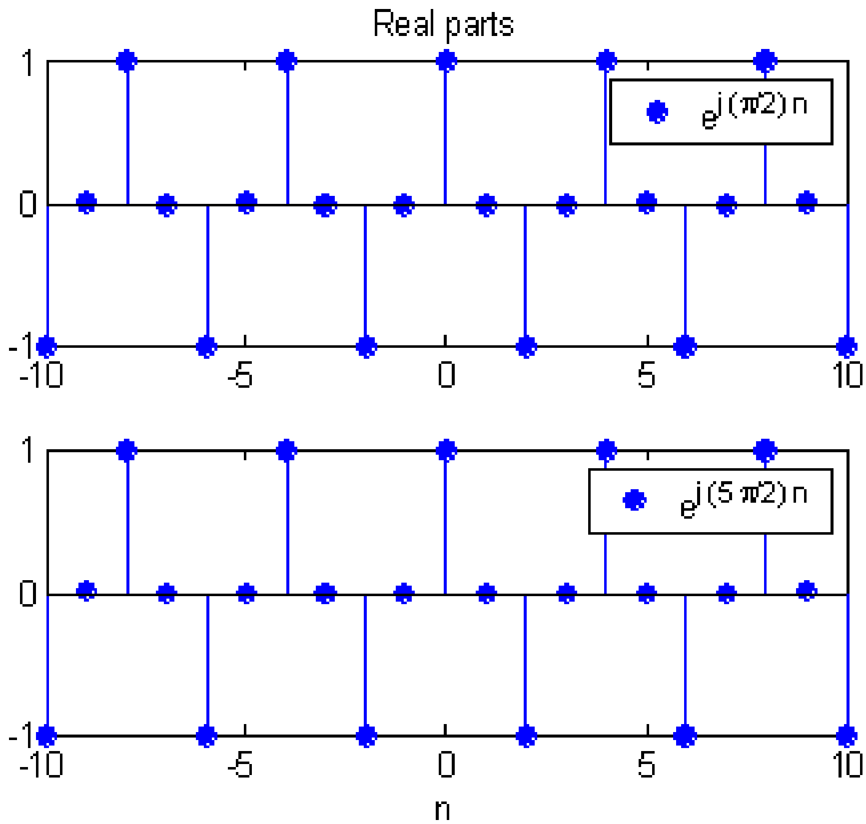


Figure 1.8: Two discrete exponentials that are identical

The rate of oscillation will increase until $\omega = \pi$, then it decreases and repeats after 2π. In Figure 1.9 we see that as we increase the angular frequency towards π the rate of oscillation increases. If you download the Matlab files included at the end of this module you can adjust the parameters and see that the rate of oscillation will decrease when exceeding π (but less than 2π).

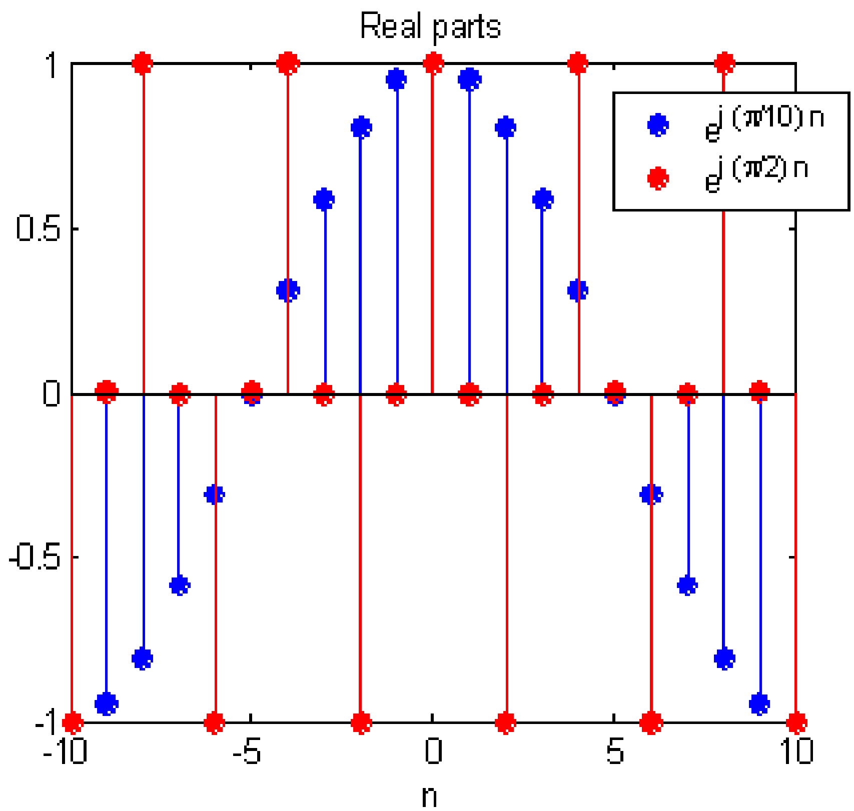


Figure 1.9: Two discrete exponentials with different frequency.

NOTE: We need to consider discrete time exponentials at an (digital angular) frequency interval of 2π only.

Low (digital angular) frequencies will correspond to ω near even multiples of π. High (digital angular) frequencies will correspond to ω near odd multiples of π.

1.4.3 Matlab files

complex_exponential.m[9]

1.4.4

Take a look at Introduction (Section 1.1); Discrete time signals (Section 1.2); Analog signals (Section 1.3); Frequency definitions and periodicity (Section 1.5); Energy & Power (Section 1.6); Exercises[10] ?

[9]http://cnx.rice.edu/content/m11527/latest/complex_exponential.m
[10]"Existence of the Minimum Variance Unbiased Estimator (MVUB)" <http://cnx.org/content/m11428/latest/>

1.5 Frequency definitions and periodicity[11]

1.5.1 Frequency definitions

In signal processing we use several types of frequencies. This may seem confusing at first, but it is really not that difficult.

1.5.1.1 Analog frequency

The **frequency** of an analog signal is the easiest to understand. A trigonometric function with argument $\Omega t = 2\pi F t$ generates a periodic function with

- a **single** frequency F.
- period T
- the relation $T = \frac{1}{F}$

Frequency is then interpreted as how many periods there are per time unit. If we choose seconds as our time unit, frequency will be measured in Hertz, which is most common.

1.5.1.2 Digital frequency

The **digital frequency** is defined as $f = \frac{F}{F_s}$, where F_s is the sampling frequency. The sampling interval is the inverse of the sampling frequency, $T_s = \frac{1}{F_s}$. A discrete time signal with **digital frequency** f therefore has a frequency given by $F = fF_s$ if the samples are spaced at $T_s = \frac{1}{F_s}$.

1.5.1.3 Consequences

In design of digital sinusoids we do not have to settle for a physical frequency. We can associate **any** physical frequency F with the digital frequency f, by choosing the appropriate sampling frequency F_s. (Using the relation $f = \frac{F}{F_s}$)

According to the relation $T_s = \frac{1}{F_s}$ choosing an appropriate sampling frequency is equivivalent to choosing a sampling interval, which implies that digital sinusoids can be designed by specifying the sampling interval.

1.5.1.4 Angular frequencies

The angular frequencies are obtained by multiplying the frequencies by the factor 2π:

Angular frequency - $\Omega = 2\pi F$
Digital angular frequency - $\omega = 2\pi f$

1.5.2 Signal periodicity

Any analog sine or cosine function is periodic. So it may seem surprising that discrete trigonometric signals not necessarily are periodic. Let us define periodicity mathematically.

If for all $k \in \mathbb{Z}$ we have

Analog signals - $x(t) = x(n + kT_0)$, then $x(t)$ is periodic with period T_0.
Discrete time signals - $x(n) = x(n + kN)$, then $x(n)$ is periodic with period N.

Example 1.3
Consider the signal $x(t) = \sin(2\pi F t)$ which obviously is periodic. You can check by using the periodicity definition and some trigonometric identitites[12] .

[11] This content is available online at <http://cnx.org/content/m11477/1.13/>.
[12] http://www.sosmath.com/trig/Trig5/trig5/trig5.html

Example 1.4
Consider the signal $x(n) = \sin(2\pi f n)$. Q:Is this signal periodic?

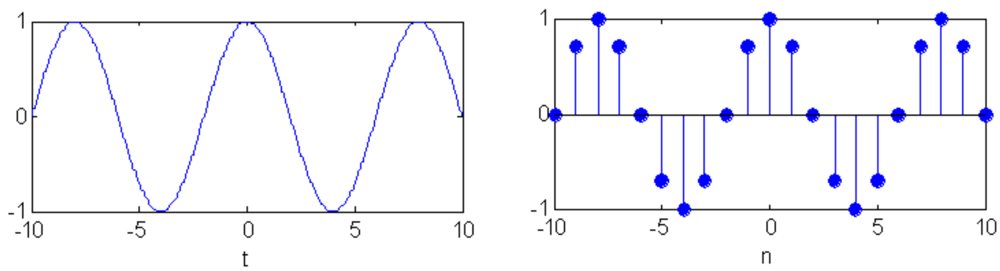
A: To check we will use the periodicity definition and some trigonometric identities[13] .

Periodicity is obtained if we can find an N which leads to $x(n) = x(n + kN)$ for all $k \in \mathbb{Z}$. Let us expand $\sin(2\pi f(n + kN))$.

$$\sin(2\pi f(n + kN)) = \sin(2\pi f n)\cos(2\pi f kN) + \cos(2\pi f n)\sin(2\pi f kN) \quad (1.5)$$

To make the right hand side of (1.5) equal to $\sin(2\pi f n)$, we need to impose a restriction on the digital frequency f. According to (1.5) only $fN = m$ will yield periodicity, $m \in \mathbb{Z}$.

Example 1.5
Consider the following signals $x(t) = \cos\left(2\pi \times \frac{1}{8}t\right)$ and $x(n) = \cos\left(2\pi \times \frac{1}{8}n\right)$, as shown in Figure 1.10.



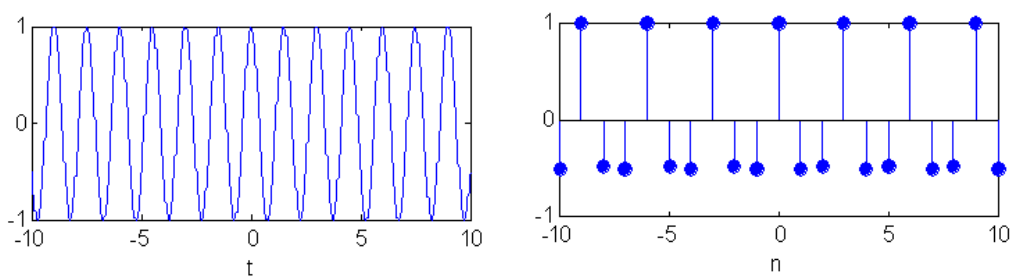
(a) (b)

Figure 1.10: (a) a) $\cos\left(2\pi \times \frac{1}{8}t\right)$ (b) b) $\cos\left(2\pi \times \frac{1}{8}n\right)$

Are the signals periodic, and if so, what are the periods?

Both the physical and digital frequency is 1/8 so both signals are periodic with period 8.

Example 1.6
Consider the following signals $x(t) = \cos\left(2\pi \times \frac{2}{3}t\right)$ and $x(n) = \cos\left(2\pi \times \frac{2}{3}n\right)$, as shown in Figure 1.11.



(a) (b)

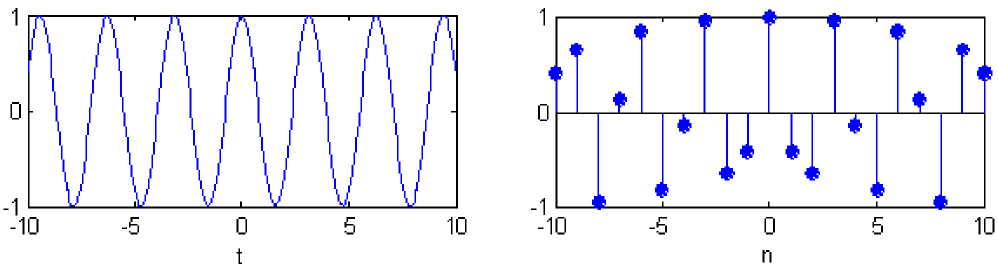
Figure 1.11: (a) a) $\cos\left(2\pi \times \frac{2}{3}t\right)$ (b) b) $\cos\left(2\pi \times \frac{2}{3}n\right)$

Are the signals periodic, and if so, what are the periods?

[13] http://www.sosmath.com/trig/Trig5/trig5/trig5.html

The frequencies are $2/3$ in both cases. The analog signal then has period $3/2$. The discrete signal has to have a period that is an integer, so the smallest possible period is then 3.

Example 1.7
Consider the following signals $x(t) = \cos(2t)$ and $x(n) = \cos(2n)$, as shown in Figure 1.12.



(a) (b)

Figure 1.12: (a) a) $\cos(2t)$ (b) b) $\cos(2n)$

Are the signals periodic, and if so, what are the periods?

The frequencies are $1/\pi$ in both cases. The analog signal then has period π. The discrete signal **is not periodic because the digital frequency is not a rational number**.

1.5.2.1 Conclusion

For a time discrete trigonometric signal to be periodic its **digital frequency has to be a rational number**, i.e. given by the ratio of two integers. Contrast this to analog trigonometric signals.

1.5.3 Matlab file

periodicity.m[14]

1.5.4

Take a look at Introduction (Section 1.1); Discrete time signals (Section 1.2); Analog signals (Section 1.3); Discrete vs Analog signals (Section 1.4); Energy & Power (Section 1.6); Exercises[15] ?

1.6 Energy and Power[16]

From physics we've learned that energy is work and power is work per time unit. Energy was measured in Joule (J) and work in Watts(W). In signal processing energy and power are defined more loosely without any necessary physical units, because the signals may represent very different physical entities. We can say that energy and power are a measure of the signal's "size".

[14]http://cnx.rice.edu/content/m11477/latest/periodicity.m
[15]"Existence of the Minimum Variance Unbiased Estimator (MVUB)" <http://cnx.org/content/m11428/latest/>
[16]This content is available online at <http://cnx.org/content/m11526/1.20/>.

1.6.1 Signal Energy

1.6.1.1 Analog signals

Since we often think of a signal as a function of varying amplitude through time, it seems to reason that a good measurement of the strength of a signal would be the area under the curve. However, this area may have a negative part. This negative part does not have less strength than a positive signal of the same size. This suggests either squaring the signal or taking its absolute value, then finding the area under that curve. It turns out that what we call the energy of a signal is the area under the squared signal, see Figure 1.13

NOTE: $E_a = \int_{-\infty}^{\infty} (|x(t)|)^2 dt$

Note that we have used squared magnitude(absolute value) if the signal should be complex valued. If the signal is real, we can leave out the magnitude operation.

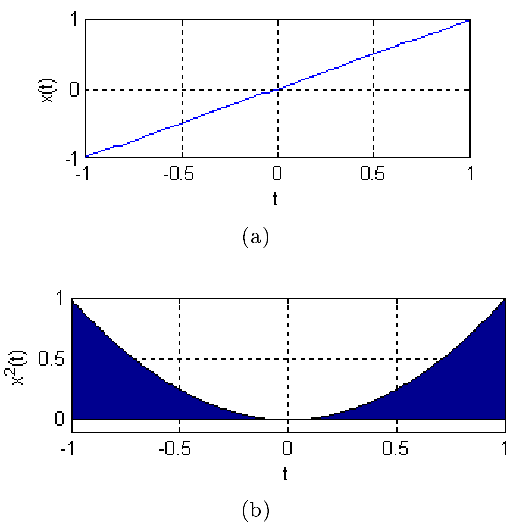


Figure 1.13: Sketch of energy calculation (a) Signal x(t) (b) The energy of x(t) is the shaded region

1.6.1.2 Discrete signals

For time discrete signals the "area under the squared signal" makes no sense, so we will have to use another energy definiton. We define energy as the sum of the squared magnitude of the samples. Mathematically

NOTE: $E_d = \sum_{n=-\infty}^{\infty} (|x(n)|)^2$

Example 1.8

Given the sequence $y(l) = b^l u(l)$, where u(l) is the unit step function. Find the energy of the sequence.

We recognize y(l) as a geometric series. Thus we can use the formula for the sum of a geometric series and we obtain the energy, $E_d = \sum_{l=0}^{\infty} (y(l))^2 = \frac{1}{1-b^2}$. This expression is only valid for $|b| < 1$. If we have a larger |b|, the series will diverge. The signal y(l) then has infinite energy. So let's have a look at power...

1.6.2 Signal Power

Our definition of energy seems reasonable, and it is. However, what if the signal does not decay fast enough? In this case we have infinite energy for any such signal. Does this mean that a fifty hertz sine wave feeding into your headphones is as strong as the fifty hertz sine wave coming out of your outlet? Obviously not. This is what leads us to the idea of **signal power**, which in such cases is a more adequate description.

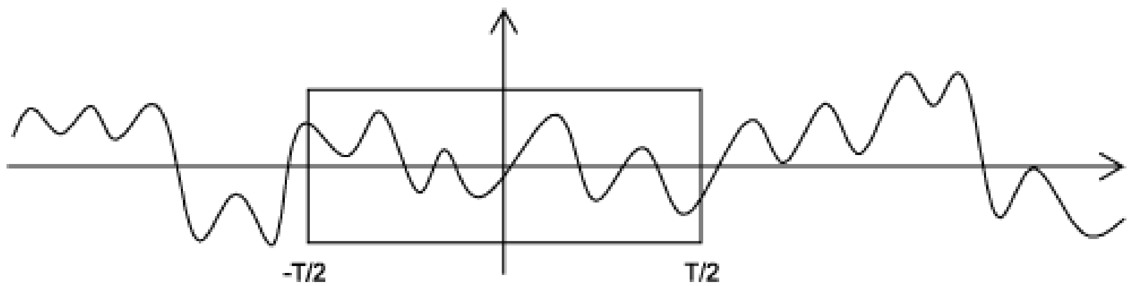


Figure 1.14: Signal with inifinite energy

1.6.2.1 Analog signals

For analog signals we define power as **energy per time interval**.

NOTE: $P_a = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} (|x(t)|)^2 dt$

1.6.2.2 Discrete signals

For time discrete signals we define power as energy per sample.

NOTE: $P_d = \frac{1}{N} \sum_{n=N_1}^{N_1+N-1} (|x(n)|)^2$

Example 1.9

Given the signals $x_1(t) = \sin(2\pi t)$ and $x_2(n) = \sin\left(\pi \frac{1}{10} n\right)$, shown in Figure 1.15, calculate the power for one period.

For the analog sine we have $P_a = \frac{1}{1} \int_0^1 \sin^2(2\pi t)\, dt = \frac{1}{2}$.

For the discrete sine we get $P_d = \frac{1}{20} \sum_{n=1}^{20} \sin^2\left(\frac{1}{10}\pi n\right) = 0.500$. Download power_sine.m[17] for plots and calculation.

[17] http://cnx.rice.edu/content/m11526/latest/power_sine.m

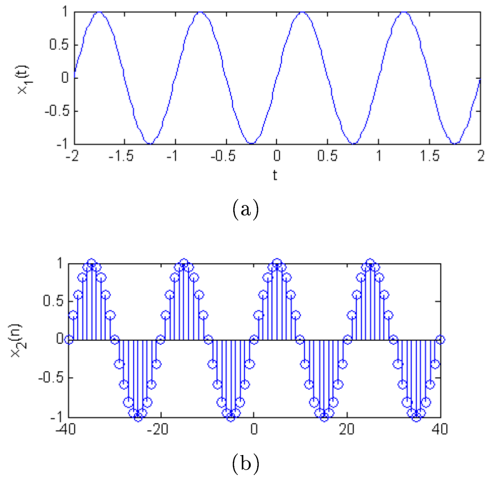


Figure 1.15: Analog and discrete time sine. (a) Analog sine (b) Discrete time sine

1.6.3 Matlab files

energy_area.m[18] power_sine.m[19]

1.6.4

Introduction (Section 1.1); Discrete time signals (Section 1.2); Analog signals (Section 1.3); Discrete vs Analog signals (Section 1.4); Frequency definitions and periodicity (Section 1.5); Exercises (Section 1.7)

1.7 Exercises[20]

Problems related to the Signals chapter. (Section 1.1)

Exercise 1.7.1 *(Solution on p. 21.)*
Find the digital frequency of $x(n) = \cos\left(2\pi \times \sqrt{3}n\right)$. Is the signal periodic? If so, find the shortest possible period.

Exercise 1.7.2 *(Solution on p. 21.)*
Find the digital frequency of $x(n) = \cos\left(2\pi \times \sqrt{4}n\right)$. Is the signal periodic? If so, find the shortest possible period.

Exercise 1.7.3 *(Solution on p. 21.)*
Find the digital frequency of $x(n) = \sin(2\pi 1.5n)$. Is the signal periodic? If so, find the shortest possible period.

Exercise 1.7.4 *(Solution on p. 21.)*
Referring to example 2 (Example 1.9) find the analog and digital frequency of $x_1(t)$ and $x_2(n)$ respectively.

---

[18] http://cnx.rice.edu/content/m11526/latest/energy_area.m
[19] http://cnx.rice.edu/content/m11526/latest/power_sine.m
[20] This content is available online at <http://cnx.org/content/m11482/1.10/>.

Solutions to Exercises in Chapter 1

Solution to Exercise 1.7.1 (p. 20)
Write $\cos\left(2\pi \times \sqrt{3}n\right)$ as $\cos\left(2\pi f n\right)$, where f is the digital frequency. We see that the digital frequency is $\sqrt{3}$. For a trigonometric signal to be periodic the digital frequency has to be a rational number, i.e $f = \frac{m}{N}$, where both m,N are integers. N is the signal period. Here the digital frequency is not a rational number, hence the signal is not periodic.

Solution to Exercise 1.7.2 (p. 20)
Write $\cos\left(2\pi \times \sqrt{4}n\right)$ as $\cos\left(2\pi f n\right)$, where f is the digital frequency. We see that the digital frequency is $\sqrt{4} = 2$. For a trigonometric signal to be periodic the digital frequency has to be a rational number, i.e $f = \frac{m}{N}$, where both m,N are integers. N is the signal period. In this case the digital frequency is a rational number, $f = \frac{2}{1}$, hence the signal is periodic. The period, N, is given by $N = \frac{m}{f} = \frac{m}{2}$. Since N has to be an integer, we obtain the shortest possible period letting $m = 2$, which yields $N = 1$.

Solution to Exercise 1.7.3 (p. 20)
Write $\sin\left(2\pi 1.5n\right)$ as $\sin\left(2\pi f n\right)$, where f is the digital frequency. We see that the digital frequency is 1.5. The digital frequency is a rational number(3/2), hence the signal is periodic. The period, N, is given by $N = \frac{m}{f} = \frac{2m}{3}$. Since N has to be an integer, we obtain the shortest possible period letting $m = 3$, which yields $N = 2$.

Solution to Exercise 1.7.4 (p. 20)
Using the same reasoning as above we easily see that the analog sine has frequency 1, while the discrete time sine has digital frequency 1/20.

Chapter 2

Convolution

2.1 Introduction to Convolution[1]

In addition to the operations performed on signals in the Signals (Section 1.2) chapter there are several more. The most important operation is linear filtering, which can be performed by **convolution**. The reason that linear filtering is so important to signal processing is that it solves many problems and that is relatively simple to describe mathematically. In this chapter we will be looking at convolution.

Convolution helps to determine the effect a system has on an input signal. It can be shown that a linear, time-invariant system[2] is completely characterized by its impulse response. Using the sampling property (Section 1.3.2.1: The (Dirac) delta function) of the delta function (Section 1.3.2.1: The (Dirac) delta function) for for continuous time signals and the unit sample (Section 1.2.3.1: The unit sample) for discrete time signals we can decompose a signal into an infinite sum / integral of scaled and shifted impulses. By knowing how a system affects a single impulse, and by understanding the way a signal is comprised of scaled and summed impulses, it seems reasonable that it should be possible to scale and sum the impulse responses of a system in order to determine what output signal will results from a particular input. This is precisely what convolution does - **convolution determines the system's output from knowledge of the input and the system's impulse response**.

Contents of this chapter

- Introduction (current module)
- Convolution - Discrete time[3]
- Convolution - Continuous time (Section 2.3)
- Properties of convolution (Section 2.5)

2.2 Discrete Time Convolution[4]

2.2.1 Introduction

Convolution, one of the most important concepts in electrical engineering, can be used to determine the output a system produces for a given input signal. It can be shown that a linear time invariant system is completely characterized by its impulse response. The sifting property of the discrete time impulse function tells us that the input signal to a system can be represented as a sum of scaled and shifted unit impulses. Thus, by linearity, it would seem reasonable to compute of the output signal as the sum of scaled and shifted

[1] This content is available online at <http://cnx.org/content/m11542/1.3/>.
[2] "System Classifications and Properties" <http://cnx.org/content/m10084/latest/>
[3] "Convolution - Discrete time" <http://cnx.org/content/m11539/latest/>
[4] This content is available online at <http://cnx.org/content/m10087/2.27/>.

unit impulse responses. That is exactly what the operation of convolution accomplishes. Hence, convolution can be used to determine a linear time invariant system's output from knowledge of the input and the impulse response.

2.2.2 Convolution and Circular Convolution

2.2.2.1 Convolution

2.2.2.1.1 Operation Definition

Discrete time convolution is an operation on two discrete time signals defined by the integral

$$(f * g)(n) = \sum_{k=-\infty}^{\infty} f(k)\, g(n-k) \tag{2.1}$$

for all signals f, g defined on $\mathbb{Z}$. It is important to note that the operation of convolution is commutative, meaning that

$$f * g = g * f \tag{2.2}$$

for all signals f, g defined on $\mathbb{Z}$. Thus, the convolution operation could have been just as easily stated using the equivalent definition

$$(f * g)(n) = \sum_{k=-\infty}^{\infty} f(n-k)\, g(k) \tag{2.3}$$

for all signals f, g defined on $\mathbb{Z}$. Convolution has several other important properties not listed here but explained and derived in a later module.

2.2.2.1.2 Definition Motivation

The above operation definition has been chosen to be particularly useful in the study of linear time invariant systems. In order to see this, consider a linear time invariant system H with unit impulse response h. Given a system input signal x we would like to compute the system output signal $H(x)$. First, we note that the input can be expressed as the convolution

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\, \delta(n-k) \tag{2.4}$$

by the sifting property of the unit impulse function. By linearity

$$Hx(n) = \sum_{k=-\infty}^{\infty} x(k)\, H\delta(n-k)\,. \tag{2.5}$$

Since $H\delta(n-k)$ is the shifted unit impulse response $h(n-k)$, this gives the result

$$Hx(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k) = (x * h)(n)\,. \tag{2.6}$$

Hence, convolution has been defined such that the output of a linear time invariant system is given by the convolution of the system input with the system unit impulse response.

2.2.2.1.3 Graphical Intuition

It is often helpful to be able to visualize the computation of a convolution in terms of graphical processes. Consider the convolution of two functions f, g given by

$$(f * g)(n) = \sum_{k=-\infty}^{\infty} f(k)\, g(n-k) = \sum_{k=-\infty}^{\infty} f(n-k)\, g(k). \tag{2.7}$$

The first step in graphically understanding the operation of convolution is to plot each of the functions. Next, one of the functions must be selected, and its plot reflected across the $k = 0$ axis. For each real t, that same function must be shifted left by t. The product of the two resulting plots is then constructed. Finally, the area under the resulting curve is computed.

Example 2.1

Recall that the impulse response for a discrete time echoing feedback system with gain a is

$$h(n) = a^n u(n), \tag{2.8}$$

and consider the response to an input signal that is another exponential

$$x(n) = b^n u(n). \tag{2.9}$$

We know that the output for this input is given by the convolution of the impulse response with the input signal

$$y(n) = x(n) * h(n). \tag{2.10}$$

We would like to compute this operation by beginning in a way that minimizes the algebraic complexity of the expression. However, in this case, each possible coice is equally simple. Thus, we would like to compute

$$y(n) = \sum_{k=-\infty}^{\infty} a^k u(k)\, b^{n-k} u(n-k). \tag{2.11}$$

The step functions can be used to further simplify this sum. Therefore,

$$y(n) = 0 \tag{2.12}$$

for $n < 0$ and

$$y(n) = \sum_{k=0}^{n} (ab)^k \tag{2.13}$$

for $n \geq 0$. Hence, provided $ab \neq 1$, we have that

$$y(n) = \{ \begin{array}{cc} 0 & n < 0 \\ \frac{1-(ab)^{n+1}}{1-(ab)} & n \geq 0 \end{array}. \tag{2.14}$$

2.2.2.2 Circular Convolution

Discrete time circular convolution is an operation on two finite length or periodic discrete time signals defined by the integral

$$(f * g)(n) = \sum_{k=0}^{N-1} \hat{f}(k)\, \hat{g}(n-k) \tag{2.15}$$

for all signals f, g defined on $\mathbb{Z}[0, N-1]$ where $\hat{f}, \hat{g}$ are periodic extensions of f and g. It is important to note that the operation of circular convolution is commutative, meaning that

$$f * g = g * f \tag{2.16}$$

for all signals f, g defined on $\mathbb{Z}[0, N-1]$. Thus, the circular convolution operation could have been just as easily stated using the equivalent definition

$$(f * g)(n) = \sum_{k=0}^{N-1} \hat{f}(n-k)\, \hat{g}(k) \tag{2.17}$$

for all signals f, g defined on $\mathbb{Z}[0, N-1]$ where $\hat{f}, \hat{g}$ are periodic extensions of f and g. Circular convolution has several other important properties not listed here but explained and derived in a later module.

Alternatively, discrete time circular convolution can be expressed as the sum of two summations given by

$$(f * g)(n) = \sum_{k=0}^{n} f(k)\, g(n-k) + \sum_{k=n+1}^{N-1} f(k)\, g(n-k+N) \tag{2.18}$$

for all signals f, g defined on $\mathbb{Z}[0, N-1]$.

Meaningful examples of computing discrete time circular convolutions in the time domain would involve complicated algebraic manipulations dealing with the wrap around behavior, which would ultimately be more confusing than helpful. Thus, none will be provided in this section. Of course, example computations in the time domain are easy to program and demonstrate. However, disrete time circular convolutions are more easily computed using frequency domain tools as will be shown in the discrete time Fourier series section.

2.2.2.2.1 Definition Motivation

The above operation definition has been chosen to be particularly useful in the study of linear time invariant systems. In order to see this, consider a linear time invariant system H with unit impulse response h. Given a finite or periodic system input signal x we would like to compute the system output signal $H(x)$. First, we note that the input can be expressed as the circular convolution

$$x(n) = \sum_{k=0}^{N-1} \hat{x}(k)\, \hat{\delta}(n-k) \tag{2.19}$$

by the sifting property of the unit impulse function. By linearity,

$$Hx(n) = \sum_{k=0}^{N-1} \hat{x}(k)\, H\, \hat{\delta}(n-k)\,. \tag{2.20}$$

Since $H\delta(n-k)$ is the shifted unit impulse response $h(n-k)$, this gives the result

$$Hx(n) = \sum_{k=0}^{N-1} \hat{x}(k)\, \hat{h}(n-k) = (x * h)(n). \tag{2.21}$$

Hence, circular convolution has been defined such that the output of a linear time invariant system is given by the convolution of the system input with the system unit impulse response.

2.2.2.2.2 Graphical Intuition

It is often helpful to be able to visualize the computation of a circular convolution in terms of graphical processes. Consider the circular convolution of two finite length functions f, g given by

$$(f * g)(n) = \sum_{k=0}^{N-1} \hat{f}(k)\, \hat{g}(n-k) = \sum_{k=0}^{N-1} \hat{f}(n-k)\, \hat{g}(k). \tag{2.22}$$

The first step in graphically understanding the operation of convolution is to plot each of the periodic extensions of the functions. Next, one of the functions must be selected, and its plot reflected across the $k=0$ axis. For each $k \in \mathbb{Z}[0, N-1]$, that same function must be shifted left by k. The product of the two resulting plots is then constructed. Finally, the area under the resulting curve on $\mathbb{Z}[0, N-1]$ is computed.

2.2.3 Interactive Element

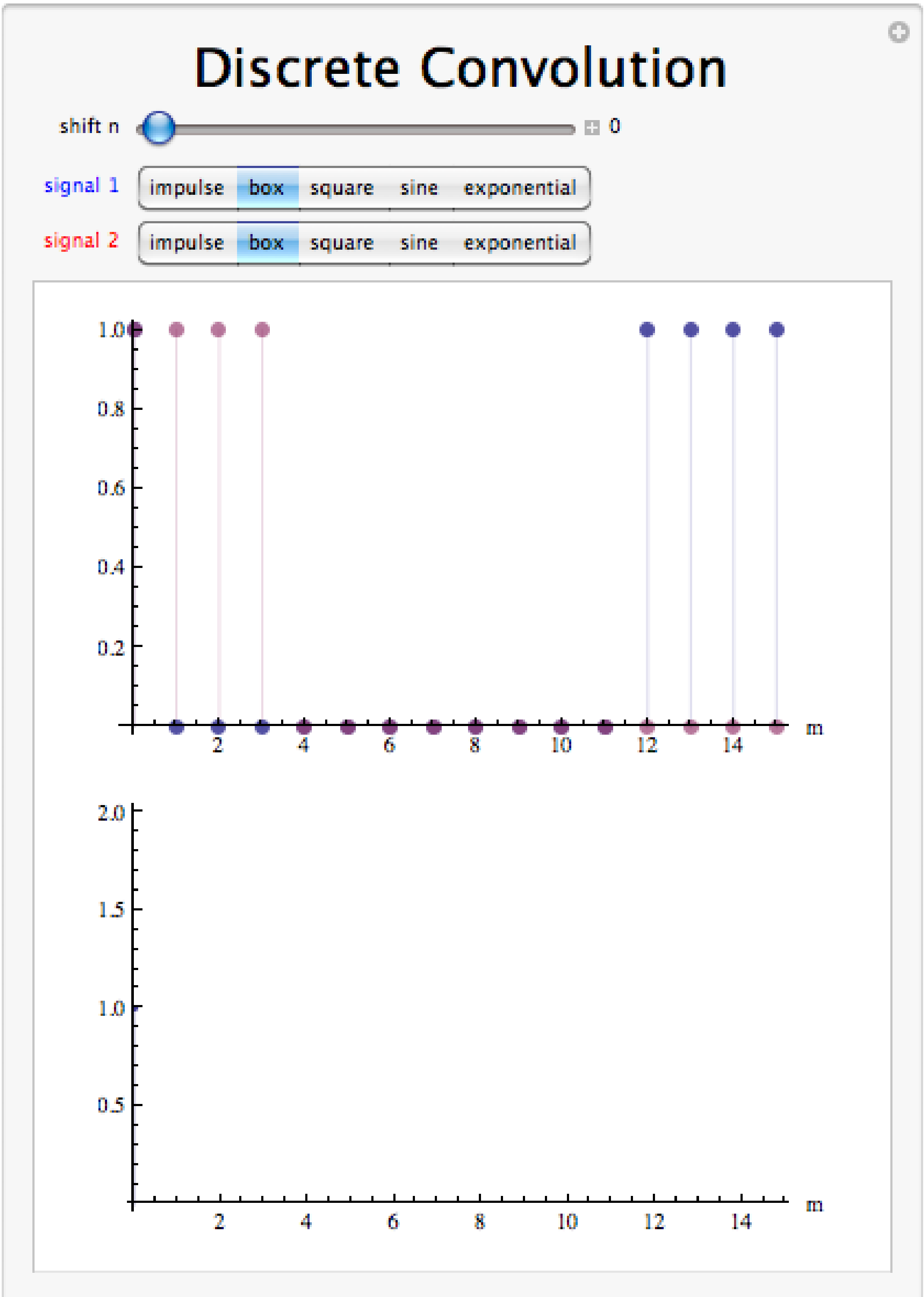


Figure 2.1: Interact (when online) with the Mathematica CDF demonstrating Discrete Linear Convolution. To download, right click and save file as .cdf

2.2.4 Convolution Summary

Convolution, one of the most important concepts in electrical engineering, can be used to determine the output signal of a linear time invariant system for a given input signal with knowledge of the system's unit impulse response. The operation of discrete time convolution is defined such that it performs this function for infinite length discrete time signals and systems. The operation of discrete time circular convolution is defined such that it performs this function for finite length and periodic discrete time signals. In each case, the output of the system is the convolution or circular convolution of the input signal with the unit impulse response.

2.3 Convolution - Analog[5]

In this module we examine convolution for continuous time signals. This will result in the convolution integral and its properties (Section 2.5). These concepts are very important in Electrical Engineering and will make any engineer's life a lot easier if the time is spent now to truly understand what is going on.

In order to fully understand convolution, you may find it useful to look at the discrete-time convolution (Section 2.2) as well. Accompanied to this module there is a fully worked out example (Section 2.4) with mathematics and figures. It will also be helpful to experiment with the Convolution - Continuous time[6] applet available from Johns Hopkins University[7] . These resources offers different approaches to this crucial concept.

2.3.1 Derivation of the convolution integral

The derivation used here closely follows the one discussed in the motivation section above. To begin this, it is necessary to state the assumptions we will be making. In this instance, the only constraints on our system are that it be linear and time-invariant.

Brief Overview of Derivation Steps:

1. An impulse input leads to an impulse response output.
2. A shifted impulse input leads to a shifted impulse response output. This is due to the time-invariance of the system.
3. We now scale the impulse input to get a scaled impulse output. This is using the scalar multiplication property of linearity.
4. We can now "sum up" an infinite number of these scaled impulses to get a sum of an infinite number of scaled impulse responses. This is using the additivity attribute of linearity.
5. Now we recognize that this infinite sum is nothing more than an integral, so we convert both sides into integrals.
6. Recognizing that the input is the function $f(t)$, we also recognize that the output is exactly the convolution integral.

[5] This content is available online at <http://cnx.org/content/m11540/1.7/>.
[6] http://www.jhu.edu/~signals/convolve/
[7] http://www.jhu.edu

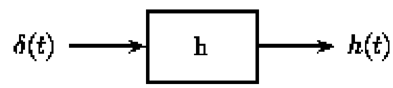


Figure 2.2: We begin with a system defined by its impulse response, $h(t)$.

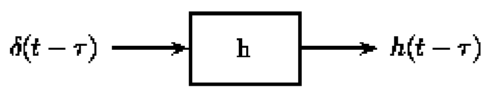


Figure 2.3: We then consider a shifted version of the input impulse. Due to the time invariance of the system, we obtain a shifted version of the output impulse response.

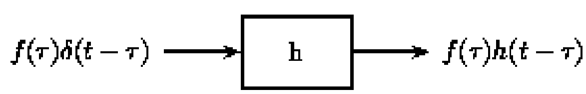


Figure 2.4: Now we use the scaling part of linearity by scaling the system by a value, $f(\tau)$, that is constant with respect to the system variable, t.

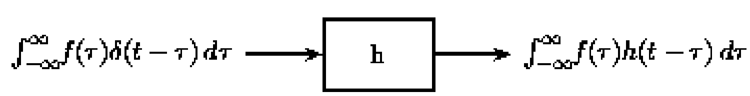


Figure 2.5: We can now use the additivity aspect of linearity to add an infinite number of these, one for each possible τ. Since an infinite sum is exactly an integral, we end up with the integration known as the Convolution Integral. Using the sampling property (Section 1.3.2.1: The (Dirac) delta function), we recognize the left-hand side simply as the input $f(t)$.

2.3.2 Convolution Integral

As mentioned above, the convolution integral provides an easy mathematical way to express the output of an LTI system based on an arbitrary signal, $x(t)$, and the system's impulse response, $h(t)$. The **convolution integral** is expressed as

$$y(t) = \int_{-\infty}^{\infty} x(\tau)\, h(t-\tau)\, d\tau \tag{2.23}$$

Convolution is such an important tool that it is represented by the symbol *, and can be written as

$$y(t) = x(t) * h(t) \tag{2.24}$$

By making a simple change of variables into the convolution integral, $\tau = t - \tau$, we can easily show that convolution is **commutative**:

$$x(t) * h(t) = h(t) * x(t) \tag{2.25}$$

which gives an equivivalent form of (2.23)

$$y(t) = \int_{-\infty}^{\infty} x(t-\tau)\, h(\tau)\, d\tau \tag{2.26}$$

For more information on the characteristics of the convolution integral, read about the Properties of Convolution (Section 2.5).

2.3.3 Implementation of Convolution

Taking a closer look at the convolution integral, we find that we are multiplying the input signal by the time-reversed impulse response and integrating. This will give us the value of the output at one given value of t. If we then shift the time-reversed impulse response by a small amount, we get the output for another value of t. Repeating this for every possible value of t, yields the total output function. While we would never actually do this computation by hand in this fashion, it does provide us with some insight into what is actually happening. We find that we are essentially reversing the impulse response function and sliding it across the input function, integrating as we go. This method, referred to as the **graphical method**, provides us with a much simpler way to solve for the output for simple (contrived) signals, while improving our intuition for the more complex cases where we rely on computers. In fact Texas Instruments[8] develops Digital Signal Processors[9] which have special instruction sets for computations such as convolution.

[8] http://www.ti.com
[9] http://dspvillage.ti.com/docs/toolssoftwarehome.jhtml

2.3.4 Summary

Convolution is a truly important concept, which **must** be well understood.

NOTE: $y(t) = \int_{-\infty}^{\infty} x(\tau) h(t-\tau) d\tau$

NOTE: $y(t) = \int_{-\infty}^{\infty} h(\tau) x(t-\tau) d\tau$

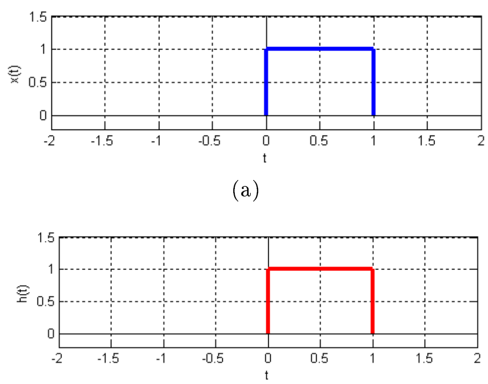
2.3.5

Go to? Introduction (Section 2.1); Convolution - Full example (Section 2.4); Convolution - Discrete time (Section 2.2); Properties of convolution (Section 2.5)

2.4 Convolution - Complete example[10]

2.4.1 Basic Example

Let us look at a basic continuous-time convolution example to help express some of the important ideas. We will convolve together two square pulses, $x(t)$ and $h(t)$, as shown in Figure 2.6



(a)

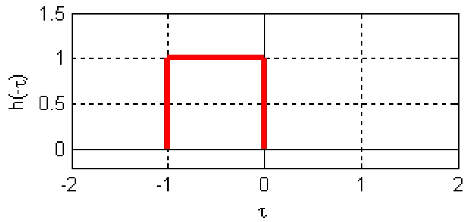
(b)

Figure 2.6: Two basic signals that we will convolve together.

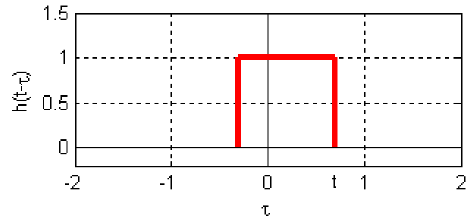
2.4.1.1 Reflect and Shift

Now we will take one of the functions and reflect it around the y-axis. Then we must shift the function, such that the origin, the point of the function that was originally on the origin, is labeled as point t. This step is shown in Figure 2.7, $h(t-\tau)$.

[10] This content is available online at <http://cnx.org/content/m11541/1.7/>.



(a)



(b)

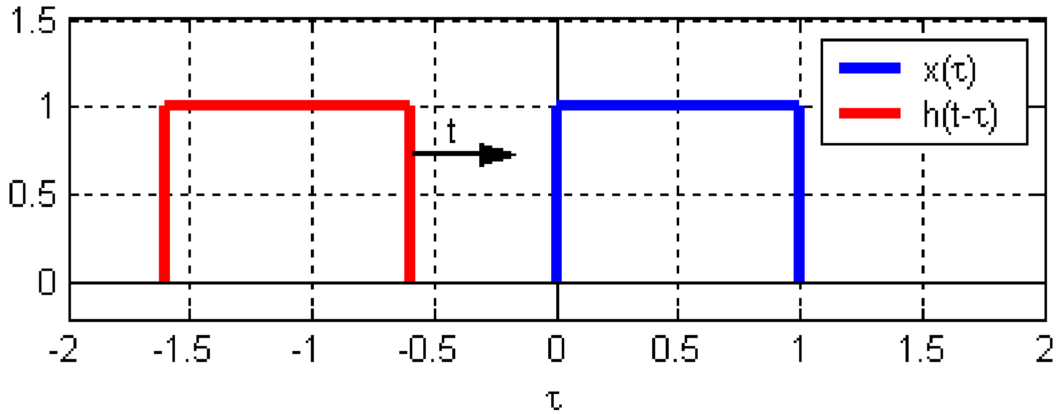
Figure 2.7: $h(-\tau)$ and $h(t-\tau)$. (a) Reflected square pulse. (b) Reflected and **shifted** square pulse.

Note that in Figure 2.7 τ is the 1st axis variable while t is a constant (in this figure). Since convolution is commutative it will never matter which function is reflected and shifted; however, as the functions become more complicated reflecting and shifting the "right one" will often make the problem much easier.

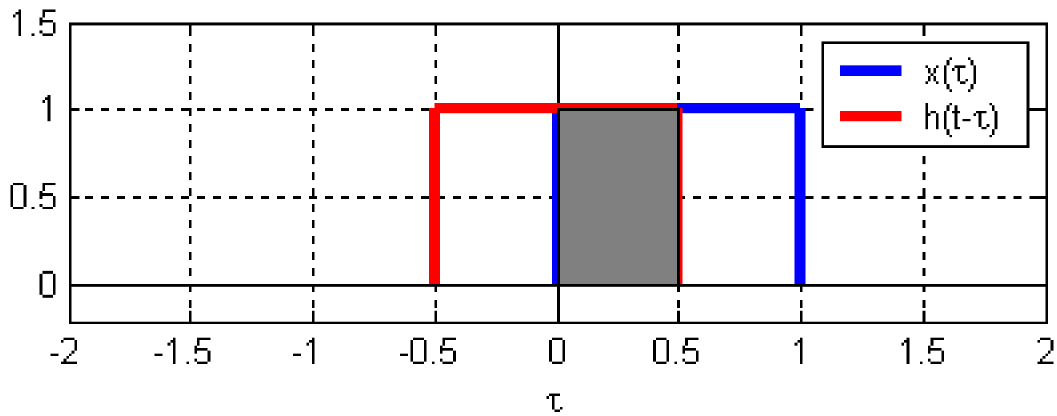
2.4.1.2 Regions of Integration

We start out with the convolution integral, $y(t) = \int_{-\infty}^{\infty} x(\tau) h(t-\tau) d\tau$. The value of the function y at time t is given by the amount of overlap(to be precise the integral of the overlapping region) between $h(t-\tau)$ and $x(\tau)$.

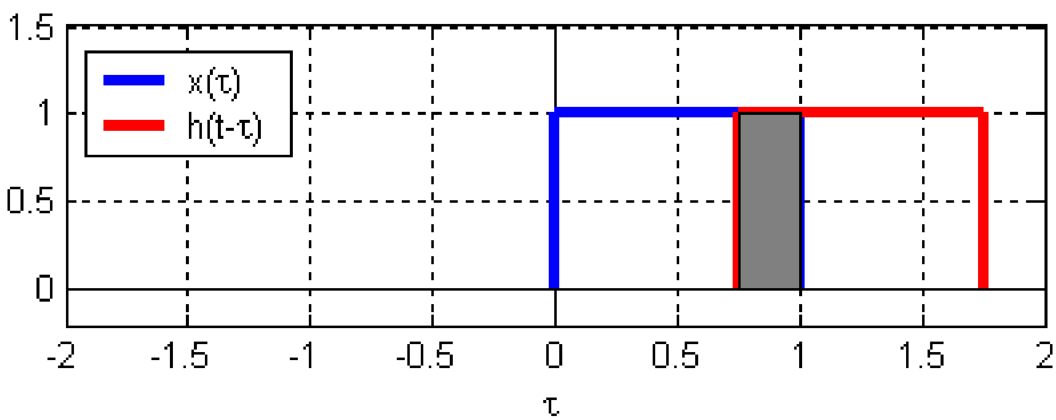
Next, we want to look at the functions and divide the span of the functions into different limits of integration. These different regions can be understood by thinking about how we slide $h(t-\tau)$ over $x(\tau)$, see Figure 2.8.



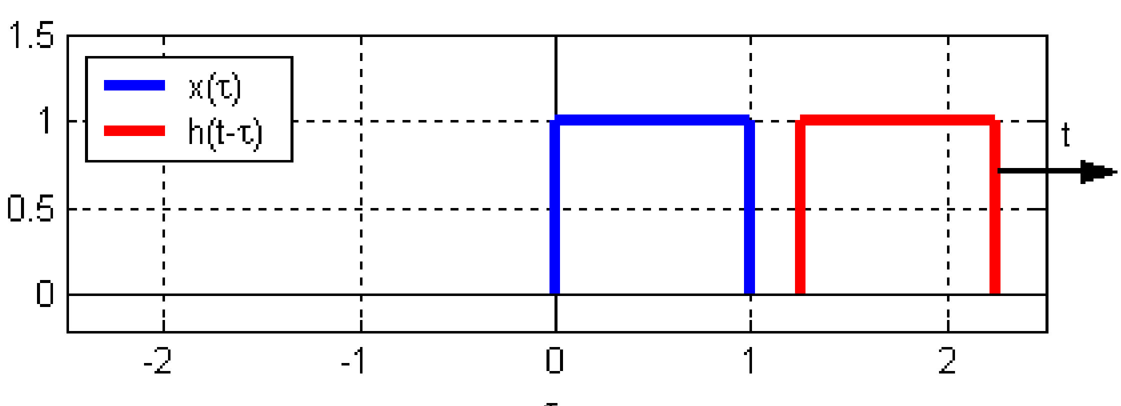
(a)



(b)



(c)



In this case we will have the following four regions. Compare these limits of integration to the four illustrations of $h(t-\tau)$ and $x(\tau)$ in Figure 2.8.

Four Limits of Integration

1. $t < 0$
2. $0 \leq t < 1$
3. $1 \leq t < 2$
4. $t \geq 2$

2.4.1.3 Using the Convolution Integral

Finally we are ready for a little math. Using the convolution integral, let us integrate the product of $x(\tau)h(t-\tau)$. For our first and fourth region this will be trivial as it will always be 0. The second region, $0 \leq t < 1$, will require the following math:

$$\begin{aligned} y(t) &= \int_0^t 1 d\tau \\ &= t \end{aligned} \tag{2.27}$$

The third region, $1 \leq t < 2$, is solved in much the same manner. Take note of the changes in our integration though. As we move $h(t-\tau)$ across our other function, the left-hand edge of the function, $t-1$, becomes our lowlimit for the integral. This is shown through our convolution integral as

$$\begin{aligned} y(t) &= \int_{t-1}^1 1 d\tau \\ &= 1-(t-1) \\ &= 2-t \end{aligned} \tag{2.28}$$

The above formulas show the method for calculating convolution; however, do not let the simplicity of this example confuse you when you work on other problems. The method will be the same, you will just have to deal with more math in more complicated integrals.

Note that the value of $y(t)$ at all time is given by the integral of the overlapping functions. In this example y for a given t equals the gray area in the plots in Figure 2.8.

2.4.1.4 Convolution Results

Thus, we have the following results for our four regions:

$$y(t) = \begin{cases} 0 \text{ if } t < 0 \\ t \text{ if } 0 \leq t < 1 \\ 2-t \text{ if } 1 \leq t < 2 \\ 0 \text{ if } t \geq 2 \end{cases} \tag{2.29}$$

Now that we have found the resulting function for each of the four regions, we can combine them together and graph the convolution of $x(t) * h(t)$.

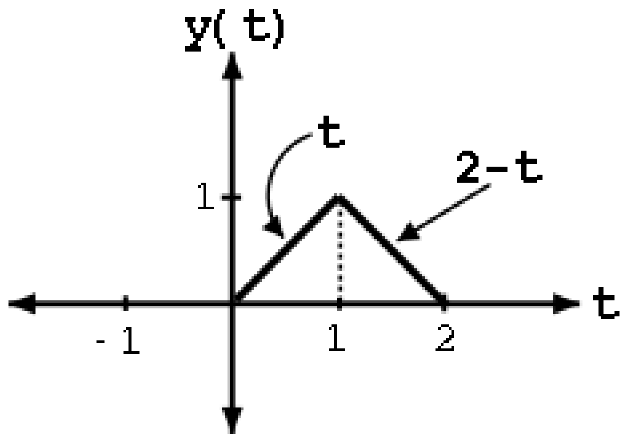


Figure 2.9: Shows the system's output in response to the input, $x(t)$.

2.4.1.5 Common sense approach

By looking at Figure 2.8 we can obtain the system output, $y(t)$, by "common" sense. For $t < 0$ there is no overlap, so $y(t)$ is 0. As t goes from 0 to 1 the overlap will **linearly** increase with a maximum for $t = 1$, the maximum corresponds to the peak value in the triangular pulse. As t goes from 1 to 2 the overlap will **linearly** decrease. For $t > 2$ there will be no overlap and hence no output.

We see readily from the "common" sense approach that the output function $y(t)$ is the same as obtained above with calculations. When convolving to square pulses the result will **always** be a triangular pulse. Its origin, peak value and strech will, of course, vary.

2.4.2

- Introduction (Section 2.1)
- Convolution - Discrete time[11]
- Convolution - Analog (Section 2.3)
- Properties of convolution (Section 2.5)

2.5 Properties of Continuous Time Convolution[12]

2.5.1 Introduction

We have already shown the important role that continuous time convolution plays in signal processing. This section provides discussion and proof of some of the important properties of continuous time convolution. Analogous properties can be shown for continuous time circular convolution with trivial modification of the proofs provided except where explicitly noted otherwise.

[11] "Convolution - Discrete time" <http://cnx.org/content/m11539/latest/>
[12] This content is available online at <http://cnx.org/content/m10088/2.17/>.

2.5.2 Continuous Time Convolution Properties

2.5.2.1 Associativity

The operation of convolution is associative. That is, for all continuous time signals f_1, f_2, f_3 the following relationship holds.

$$f_1 * (f_2 * f_3) = (f_1 * f_2) * f_3 \tag{2.30}$$

In order to show this, note that

$$\begin{aligned} (f_1 * (f_2 * f_3))(t) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(\tau_1) f_2(\tau_2) f_3((t - \tau_1) - \tau_2)\, d\tau_2 d\tau_1 \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(\tau_1) f_2((\tau_1 + \tau_2) - \tau_1) f_3(t - (\tau_1 + \tau_2))\, d\tau_2 d\tau_1 \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(\tau_1) f_2(\tau_3 - \tau_1) f_3(t - \tau_3)\, d\tau_1 d\tau_3 \\ &= ((f_1 * f_2) * f_3)(t) \end{aligned} \tag{2.31}$$

proving the relationship as desired through the substitution $\tau_3 = \tau_1 + \tau_2$.

2.5.2.2 Commutativity

The operation of convolution is commutative. That is, for all continuous time signals f_1, f_2 the following relationship holds.

$$f_1 * f_2 = f_2 * f_1 \tag{2.32}$$

In order to show this, note that

$$\begin{aligned} (f_1 * f_2)(t) &= \int_{-\infty}^{\infty} f_1(\tau_1) f_2(t - \tau_1)\, d\tau_1 \\ &= \int_{-\infty}^{\infty} f_1(t - \tau_2) f_2(\tau_2)\, d\tau_2 \\ &= (f_2 * f_1)(t) \end{aligned} \tag{2.33}$$

proving the relationship as desired through the substitution $\tau_2 = t - \tau_1$.

2.5.2.3 Distribitivity

The operation of convolution is distributive over the operation of addition. That is, for all continuous time signals f_1, f_2, f_3 the following relationship holds.

$$f_1 * (f_2 + f_3) = f_1 * f_2 + f_1 * f_3 \tag{2.34}$$

In order to show this, note that

$$\begin{aligned} (f_1 * (f_2 + f_3))(t) &= \int_{-\infty}^{\infty} f_1(\tau) (f_2(t - \tau) + f_3(t - \tau))\, d\tau \\ &= \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau)\, d\tau + \int_{-\infty}^{\infty} f_1(\tau) f_3(t - \tau)\, d\tau \\ &= (f_1 * f_2 + f_1 * f_3)(t) \end{aligned} \tag{2.35}$$

proving the relationship as desired.

2.5.2.4 Multilinearity

The operation of convolution is linear in each of the two function variables. Additivity in each variable results from distributivity of convolution over addition. Homogenity of order one in each varible results from the fact that for all continuous time signals f_1, f_2 and scalars a the following relationship holds.

$$a(f_1 * f_2) = (af_1) * f_2 = f_1 * (af_2) \tag{2.36}$$

In order to show this, note that

$$\begin{aligned} (a(f_1 * f_2))(t) &= a\int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau)\, d\tau \\ &= \int_{-\infty}^{\infty} (af_1(\tau)) f_2(t-\tau)\, d\tau \\ &= ((af_1) * f_2)(t) \\ &= \int_{-\infty}^{\infty} f_1(\tau)(af_2(t-\tau))\, d\tau \\ &= (f_1 * (af_2))(t) \end{aligned} \tag{2.37}$$

proving the relationship as desired.

2.5.2.5 Conjugation

The operation of convolution has the following property for all continuous time signals f_1, f_2.

$$\overline{f_1 * f_2} = \overline{f_1} * \overline{f_2} \tag{2.38}$$

In order to show this, note that

$$\begin{aligned} \left(\overline{f_1 * f_2}\right)(t) &= \overline{\int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau)\, d\tau} \\ &= \int_{-\infty}^{\infty} \overline{f_1(\tau) f_2(t-\tau)} d\tau \\ &= \int_{-\infty}^{\infty} \overline{f_1}(\tau)\, \overline{f_2}(t-\tau)\, d\tau \\ &= \left(\overline{f_1} * \overline{f_2}\right)(t) \end{aligned} \tag{2.39}$$

proving the relationship as desired.

2.5.2.6 Time Shift

The operation of convolution has the following property for all continuous time signals f_1, f_2 where S_T is the time shift operator.

$$S_T(f_1 * f_2) = (S_T f_1) * f_2 = f_1 * (S_T f_2) \tag{2.40}$$

In order to show this, note that

$$\begin{aligned} S_T(f_1 * f_2)(t) &= \int_{-\infty}^{\infty} f_2(\tau) f_1((t-T)-\tau)\, d\tau \\ &= \int_{-\infty}^{\infty} f_2(\tau) S_T f_1(t-\tau)\, d\tau \\ &= ((S_T f_1) * f_2)(t) \\ &= \int_{-\infty}^{\infty} f_1(\tau) f_2((t-T)-\tau)\, d\tau \\ &= \int_{-\infty}^{\infty} f_1(\tau) S_T f_2(t-\tau)\, d\tau \\ &= f_1 * (S_T f_2)(t) \end{aligned} \tag{2.41}$$

proving the relationship as desired.

2.5.2.7 Differentiation

The operation of convolution has the following property for all continuous time signals f_1, f_2.

$$\frac{d}{dt}(f_1 * f_2)(t) = \left(\frac{df_1}{dt} * f_2\right)(t) = \left(f_1 * \frac{df_2}{dt}\right)(t) \tag{2.42}$$

In order to show this, note that

$$\begin{aligned} \frac{d}{dt}(f_1 * f_2)(t) &= \int_{-\infty}^{\infty} f_2(\tau) \frac{d}{dt} f_1(t-\tau)\, d\tau \\ &= \left(\frac{df_1}{dt} * f_2\right)(t) \\ &= \int_{-\infty}^{\infty} f_1(\tau) \frac{d}{dt} f_2(t-\tau)\, d\tau \\ &= \left(f_1 * \frac{df_2}{dt}\right)(t) \end{aligned} \tag{2.43}$$

proving the relationship as desired.

2.5.2.8 Impulse Convolution

The operation of convolution has the following property for all continuous time signals f where δ is the Dirac delta funciton.

$$f * \delta = f \tag{2.44}$$

In order to show this, note that

$$\begin{aligned} (f * \delta)(t) &= \int_{-\infty}^{\infty} f(\tau)\, \delta(t-\tau)\, d\tau \\ &= f(t) \int_{-\infty}^{\infty} \delta(t-\tau)\, d\tau \\ &= f(t) \end{aligned} \tag{2.45}$$

proving the relationship as desired.

2.5.2.9 Width

The operation of convolution has the following property for all continuous time signals f_1, f_2 where $Duration(f)$ gives the duration of a signal f.

$$Duration(f_1 * f_2) = Duration(f_1) + Duration(f_2) \tag{2.46}$$

. In order to show this informally, note that $(f_1 * f_2)(t)$ is nonzero for all t for which there is a τ such that $f_1(\tau) f_2(t-\tau)$ is nonzero. When viewing one function as reversed and sliding past the other, it is easy to see that such a τ exists for all t on an interval of length $Duration(f_1) + Duration(f_2)$. Note that this is not always true of circular convolution of finite length and periodic signals as there is then a maximum possible duration within a period.

2.5.3 Convolution Properties Summary

As can be seen the operation of continuous time convolution has several important properties that have been listed and proven in this module. With slight modifications to proofs, most of these also extend to continuous time circular convolution as well and the cases in which exceptions occur have been noted above. These identities will be useful to keep in mind as the reader continues to study signals and systems.

Chapter 3

Analog Filtering

3.1 Frequency response from a circuit diagram[1]

In this module we calculate the **frequency response** from a circuit diagram of a simple analog filter, as shown in Figure 3.1 (Simple Circuit). We know that the frequency response, denoted by $H(j(\Omega))$, is calculated as ratio of the output and input voltages (in the frequency domain). That is,

$$\frac{V_{\text{out}}}{V_{\text{in}}} = H(j\Omega) \tag{3.1}$$

Notice that we use capital letters in these relations. This is to indicate that they are frequency domain descriptions.

Now, to calculate the frequency response we find expressions for V_{in}, and V_{out}, as follows

$$V_{\text{in}} = IR + V_{\text{out}} \tag{3.2}$$

Further, the current in the circuit can be expressed as

$$I = jC\Omega V_{\text{out}} \tag{3.3}$$

Then, the frequency response is given as:

$$\begin{aligned} \frac{V_{\text{out}}}{V_{\text{in}}} &= H(j\Omega) \\ &= \frac{1}{j\Omega RC+1} \end{aligned} \tag{3.4}$$

Note that above we have used **impedance** considerations. Have a look at The Impedance concept[2] and Impedance[3] for a quick summary of impedance considerations.

Implicit in using the transfer function is that the input is a complex exponential, and the output is also a complex exponential having the same frequency. The transfer function reveals how the circuit modifies the input amplitude in creating the output amplitude. Thus, the transfer function **completely** describes how the circuit processes the input complex exponential to produce the output complex exponential. The circuit's function is thus summarized by the transfer function. In fact, circuits are often designed to meet transfer function specifications. Because transfer functions are complex-valued, frequency-dependent quantities, we can better appreciate a circuit's function by examining the magnitude and phase of its transfer function (Figure 3.2 (Magnitude and phase of the transfer function)). Note that in Figure 3.2 (Magnitude and phase of the transfer function) we plot the magnitude phase as a function of the frequency F, instead of the

[1]This content is available online at <http://cnx.org/content/m13646/1.2/>.
[2]"The Impedance Concept" <http://cnx.org/content/m0024/latest/>
[3]"Impedance" <http://cnx.org/content/m0025/latest/>

angular frequency Ω. Since $\Omega = 2\pi F$, this is just a matter of taste, see Frequency definitions and peridocity (Section 1.5) for details.

Simple Circuit

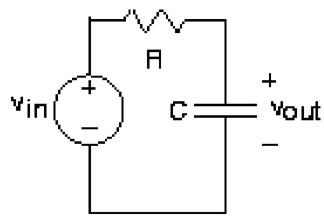


Figure 3.1: A simple RC circuit.

Magnitude and phase of the transfer function

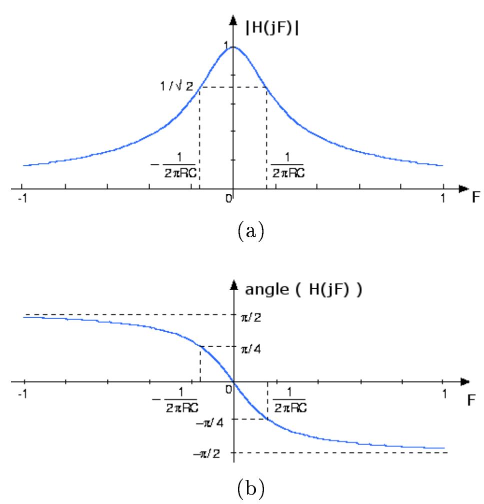


Figure 3.2: Magnitude and phase of the transfer function of the RC circuit shown in Figure 3.1 (Simple Circuit) when $RC = 1$. (a) $|H(jF)| = \frac{1}{\sqrt{(2\pi FRC)^2+1}}$ (b) $\angle(H(jF)) = -\arctan(2\pi FRC)$

Several things to note about this transfer function.

We can compute the frequency response for both positive and negative frequencies. Recall that sinusoids consist of the sum of two complex exponentials, one having the negative frequency of the other. We will consider how the circuit acts on a sinusoid soon. Do note that the magnitude has **even symmetry**: The negative frequency portion is a mirror image of the positive frequency portion: $|H(-(jF))| = |H(jF)|$. The phase has **odd symmetry**: $\angle(H(-(jF))) = -\angle(H(jF))$. These properties of this specific example apply for **all** transfer functions associated with circuits. Consequently, we don't need to plot the negative frequency component; we know what it is from the positive frequency part.

The magnitude equals $\frac{1}{\sqrt{2}}$ of its maximum gain (1 at $F = 0$) when $2\pi FRC = 1$ (the two terms in the denominator of the magnitude are equal). The frequency $F_c = \frac{1}{2\pi RC}$ defines the boundary between two operating ranges.

- For frequencies below this frequency, the circuit does not much alter the amplitude of the complex exponential source.
- For frequencies greater than F_c, the circuit strongly attenuates the amplitude. Thus, when the source frequency is in this range, the circuit's output has a much smaller amplitude than that of the source.

For these reasons, this frequency is known as the **cutoff frequency**. In this circuit the cutoff frequency depends **only** on the product of the resistance and the capacitance. Thus, a cutoff frequency of 1 kHz occurs when $\frac{1}{2\pi RC} = 10^3$ or $RC = \frac{10^{-3}}{2\pi} = 1.59 \times 10^{-4}$. Thus resistance-capacitance combinations of 1.59 kΩ and 100 nF or 10 Ω and 1.59 μF result in the **same** cutoff frequency.

The phase shift caused by the circuit at the cutoff frequency precisely equals $-\frac{\pi}{4}$. Thus, below the cutoff frequency, phase is little affected, but at higher frequencies, the phase shift caused by the circuit becomes $-\frac{\pi}{2}$. This phase shift corresponds to the difference between a cosine and a sine.

We can use the transfer function to find the output when the input voltage is a sinusoid for two reasons. First of all, a sinusoid is the sum of two complex exponentials, each having a frequency equal to the negative of the other. Secondly, because the circuit is linear, superposition applies. If the source is a sine wave, we know that

$$\begin{aligned} v_{\text{in}}(t) &= A\sin(\Omega t) \\ &= \frac{A}{2j}\left(e^{j\Omega t} - e^{-(j\Omega t)}\right) \end{aligned} \tag{3.5}$$

Since the input is the sum of two complex exponentials, we know that the output is also a sum of two similar complex exponentials, the only difference being that the complex amplitude of each is multiplied by the transfer function evaluated at each exponential's frequency.

$$v_{\text{out}}(t) = \frac{A}{2j}H(j\Omega)e^{j\Omega t} - \frac{A}{2j}H(-(j\Omega))e^{-(j\Omega t)} \tag{3.6}$$

As noted earlier, the transfer function is most conveniently expressed in polar form: $H(j\Omega) = |H(j\Omega)|e^{j\angle(H(j\Omega))}$. Furthermore, $|H(-(j\Omega))| = |H(j\Omega)|$ (even symmetry of the magnitude) and $\angle(H(-(j\Omega))) = -\angle(H(j\Omega))$ (odd symmetry of the phase). The output voltage expression simplifies to

$$\begin{aligned} v_{\text{out}}(t) &= A|H(j\Omega)|\sin(\Omega t + \angle(H(j\Omega))) \\ &= \frac{A}{2j}|H(j(\Omega))|e^{j\Omega t + \angle(H(j\Omega))} - \frac{A}{2j}|H(j\Omega)|e^{(-(j\Omega t)) - \angle(H(j\Omega))} \end{aligned} \tag{3.7}$$

The circuit's output to a sinusoidal input is also a sinusoid, having a gain equal to the magnitude of the circuit's transfer function evaluated at the source frequency and a phase equal to the phase of the transfer function at the source frequency. It will turn out that this input-output relation description applies to any linear circuit having a sinusoidal source.

The notion of impedance arises when we assume the sources are complex exponentials. This assumption may seem restrictive; what would we do if the source were a unit step? When we use impedances to find the transfer function between the source and the output variable, we can derive from it the differential equation that relates input and output. The differential equation applies no matter what the source may be. As we have argued, it is far simpler to use impedances to find the differential equation (because we can use series and parallel combination rules) than any other method. In this sense, we have not lost anything by temporarily pretending the source is a complex exponential.

In fact we can also solve the differential equation using impedances! Thus, despite the apparent restrictiveness of impedances, assuming complex exponential sources is actually quite general.

Chapter 4

Sampling

4.1 Introduction[1]

Contents of Sampling chapter

4.1.1 Why sample?

This section introduces sampling. Sampling is the necessary fundament for all digital signal processing and communication. Sampling can be defined as the process of measuring an analog signal at distinct points.

Digital representation of analog signals offers advantages in terms of

- robustness towards noise, meaning we can send more bits/s
- use of flexible processing equipment, in particular the computer
- more reliable processing equipment
- easier to adapt complex algorithms

[1]This content is available online at <http://cnx.org/content/m11419/1.29/>.

4.1.2 Claude E. Shannon

Figure 4.1: Claude Elwood Shannon (1916-2001)

Claude Shannon[2] has been called the father of information theory, mainly due to his landmark papers on the "Mathematical theory of communication"[3] . Harry Nyquist[4] was the first to state the sampling theorem in 1928, but it was not proven until Shannon proved it 21 years later in the paper "Communications in the presence of noise"[5] .

4.1.3 Notation

In this chapter we will be using the following notation

- Original analog signal $x(t)$
- Sampling frequency F_s
- Sampling interval T_s (Note that: $F_s = \frac{1}{T_s}$)
- Sampled signal $x_s(n)$. (Note that $x_s(n) = x(nT_s)$)
- Real angular frequency Ω
- Digital angular frequency ω. (Note that: $\omega = \Omega T_s$)

4.1.4 The Sampling Theorem

NOTE: When sampling an analog signal the sampling frequency must be greater than twice the highest frequency component of the analog signal to be able to reconstruct the original signal from the sampled version.

[2] http://www.research.att.com/~njas/doc/ces5.html
[3] http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf
[4] http://www.wikipedia.org/wiki/Harry_Nyquist
[5] http://www.stanford.edu/class/ee104/shannonpaper.pdf

4.1.5

Finished? Have at look at: Proof (Section 4.2); Illustrations (Section 4.3); Matlab Example (Section 4.4); Aliasing applet (Section 4.5); Hold operation (Section 4.6); System view (Section 4.7); Exercises (Section 4.8)

4.2 Proof[6]

NOTE: In order to recover the signal $x(t)$ from it's samples exactly, it is necessary to sample $x(t)$ at a rate greater than twice it's highest frequency component.

4.2.1 Introduction

As mentioned earlier (p. 45), sampling is the necessary fundament when we want to apply digital signal processing on analog signals.

Here we present the proof of the sampling theorem. The proof is divided in two. First we find an expression for the spectrum of the signal resulting from sampling the original signal $x(t)$. Next we show that the signal $x(t)$ can be recovered from the samples. Often it is easier using the frequency domain when carrying out a proof, and this is also the case here.

Key points in the proof

- We find an equation (4.8) for the spectrum of the sampled signal
- We find a simple method to reconstruct (4.14) the original signal
- The sampled signal has a periodic spectrum...
- ...and the period is $2 \times \pi F_s$

4.2.2 Proof part 1 - Spectral considerations

By sampling $x(t)$ every T_s second we obtain $x_s(n)$. The inverse fourier transform of this time discrete signal (Section 1.2) is

$$x_s(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X_s\left(e^{j\omega}\right) e^{j\omega n} d\omega \tag{4.1}$$

For convenience we express the equation in terms of the real angular frequency Ω using $\omega = \Omega T_s$. We then obtain

$$x_s(n) = \frac{T_s}{2\pi} \int_{\frac{-\pi}{T_s}}^{\frac{\pi}{T_s}} X_s\left(e^{j\Omega T_s}\right) e^{j\Omega T_s n} d\Omega \tag{4.2}$$

The inverse fourier transform of a continuous signal is

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega) e^{j\Omega t} d\Omega \tag{4.3}$$

From this equation we find an expression for $x(nT_s)$

$$x(nT_s) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega) e^{j\Omega n T_s} d\Omega \tag{4.4}$$

To account for the difference in region of integration we split the integration in (4.4) into subintervals of length $\frac{2\pi}{T_s}$ and then take the sum over the resulting integrals to obtain the complete area.

$$x(nT_s) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \int_{\frac{(2k-1)\pi}{T_s}}^{\frac{(2k+1)\pi}{T_s}} X(j\Omega) e^{j\Omega n T_s} d\Omega \tag{4.5}$$

[6] This content is available online at <http://cnx.org/content/m11423/1.27/>.

Then we change the integration variable, setting $\Omega = \eta + \frac{2\times\pi k}{T_s}$

$$x(nT_s) = \frac{1}{2\pi}\sum_{k=-\infty}^{\infty}\int_{\frac{-\pi}{T_s}}^{\frac{\pi}{T_s}} X\left(j\left(\eta + \frac{2\times\pi k}{T_s}\right)\right) e^{j\left(\eta+\frac{2\times\pi k}{T_s}\right)nT_s} d\eta \tag{4.6}$$

We obtain the final form by observing that $e^{j2\times\pi kn} = 1$, reinserting $\eta = \Omega$ and multiplying by $\frac{T_s}{T_s}$

$$x(nT_s) = \frac{T_s}{2\pi}\int_{\frac{-\pi}{T_s}}^{\frac{\pi}{T_s}} \sum_{k=-\infty}^{\infty}\frac{1}{T_s} X\left(j\left(\Omega + \frac{2\times\pi k}{T_s}\right)\right) e^{j\Omega nT_s} d\Omega \tag{4.7}$$

To make $x_s(n) = x(nT_s)$ for all values of n, the integrands in (4.2) and (4.7) have to agreee, that is

$$X_s\left(e^{j\Omega T_s}\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X\left(j\left(\Omega + \frac{2\pi k}{T_s}\right)\right) \tag{4.8}$$

This is a central result. We see that the digital spectrum consists of a sum of shifted versions of the original, analog spectrum. Observe the periodicity!

We can also express this relation in terms of the digital angular frequency $\omega = \Omega T_s$

$$X_s\left(e^{j\omega}\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X\left(j\frac{\omega + 2\times\pi k}{T_s}\right) \tag{4.9}$$

This concludes the first part of the proof. Now we want to find a reconstruction formula, so that we can recover $x(t)$ from $x_s(n)$.

4.2.3 Proof part II - Signal reconstruction

For a bandlimited (Figure 4.3) signal the inverse fourier transform is

$$x(t) = \frac{1}{2\pi}\int_{\frac{-\pi}{T_s}}^{\frac{\pi}{T_s}} X(j\Omega)\, e^{j\Omega t} d\Omega \tag{4.10}$$

In the interval we are integrating we have: $X_s\left(e^{j\Omega T_s}\right) = \frac{X(j\Omega)}{T_s}$. Substituting this relation into (4.10) we get

$$x(t) = \frac{T_s}{2\pi}\int_{\frac{-\pi}{T_s}}^{\frac{\pi}{T_s}} X_s\left(e^{j\Omega T_s}\right) e^{j\Omega t} d\Omega \tag{4.11}$$

Using the DTFT (Chapter 8) relation for $X_s\left(e^{j\Omega T_s}\right)$ we have

$$x(t) = \frac{T_s}{2\pi}\int_{\frac{-\pi}{T_s}}^{\frac{\pi}{T_s}} \sum_{n=-\infty}^{\infty} x_s(n)\, e^{-(j\Omega nT_s)} e^{j\Omega t} d\Omega \tag{4.12}$$

Interchanging integration and summation (under the assumption of convergence) leads to

$$x(t) = \frac{T_s}{2\pi}\sum_{n=-\infty}^{\infty} x_s(n)\int_{\frac{-\pi}{T_s}}^{\frac{\pi}{T_s}} e^{j\Omega(t-nT_s)} d\Omega \tag{4.13}$$

Finally we perform the integration and arrive at the important reconstruction formula

$$x(t) = \sum_{n=-\infty}^{\infty} x_s(n)\,\frac{\sin\left(\frac{\pi}{T_s}(t-nT_s)\right)}{\frac{\pi}{T_s}(t-nT_s)} \tag{4.14}$$

(Thanks to R.Loos for pointing out an error in the proof.)

4.2.4 Summary

NOTE: $X_s\left(e^{j\Omega T_s}\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X\left(j\left(\Omega + \frac{2\pi k}{T_s}\right)\right)$

NOTE: $x(t) = \sum_{n=-\infty}^{\infty} x_s(n) \frac{\sin\left(\frac{\pi}{T_s}(t-nT_s)\right)}{\frac{\pi}{T_s}(t-nT_s)}$

4.2.5

Go to Introduction (Section 4.1); Illustrations (Section 4.3); Matlab Example (Section 4.4); Hold operation (Section 4.6); Aliasing applet (Section 4.5); System view (Section 4.7); Exercises (Section 4.8) ?

4.3 Illustrations[7]

In this module we illustrate the processes involved in sampling and reconstruction. To see how all these processes work together as a whole, take a look at the system view (Section 4.7). In Sampling and reconstruction with Matlab (Section 4.4) we provide a Matlab script for download. The matlab script shows the process of sampling and reconstruction **live**.

4.3.1 Basic examples

Example 4.1
To sample an analog signal with 3000 Hz as the highest frequency component requires sampling at 6000 Hz or above.

Example 4.2
The sampling theorem can also be applied in two dimensions, i.e. for image analysis. A 2D sampling theorem has a simple physical interpretation in image analysis: Choose the sampling interval such that it is less than or equal to half of the smallest interesting detail in the image.

4.3.2 The process of sampling

We start off with an analog signal. This can for example be the sound coming from your stereo at home or your friend talking.

The signal is then sampled uniformly. Uniform sampling implies that we sample every T_s seconds. In Figure 4.2 we see an analog signal. The analog signal has been sampled at times $t = nT_s$.

[7]This content is available online at <http://cnx.org/content/m11443/1.33/>.

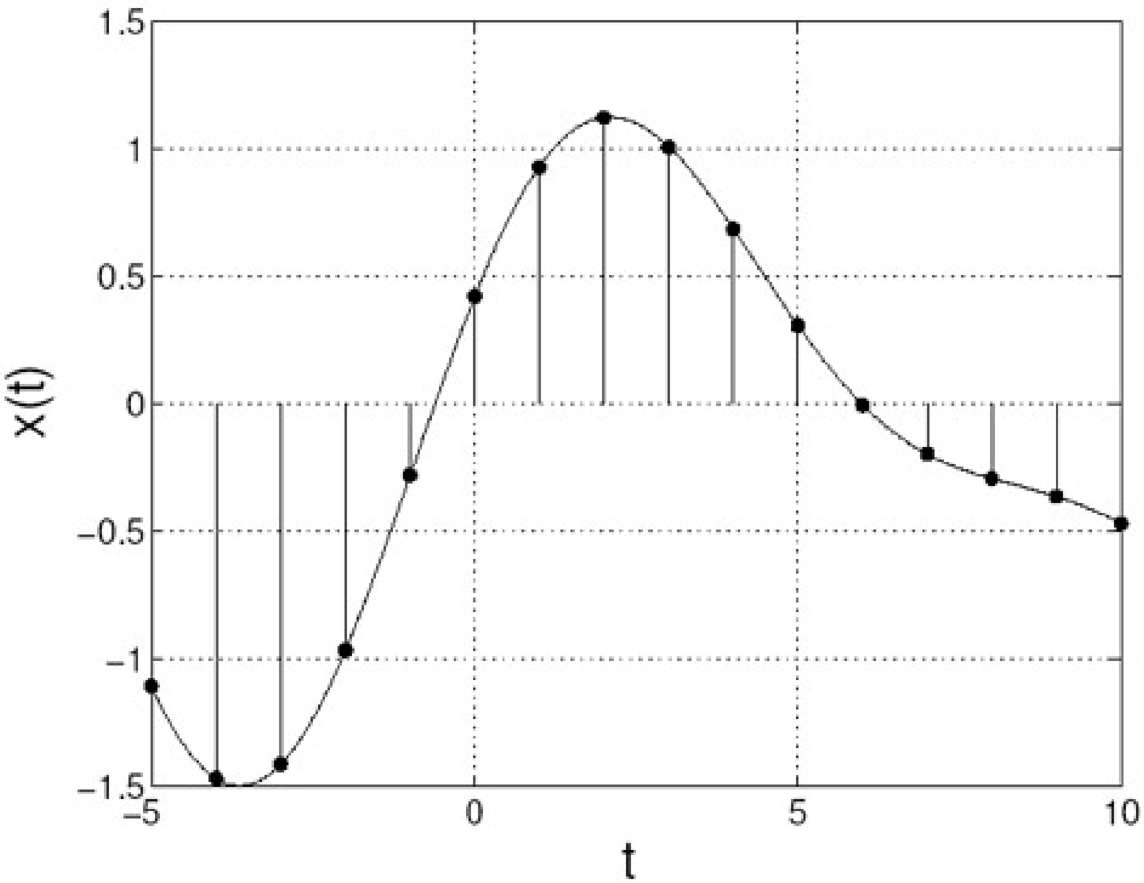


Figure 4.2: Analog signal, samples are marked with dots.

In signal processing it is often more convenient and easier to work in the frequency domain. So let's look at at the signal in frequency domain, Figure 4.3. For illustration purposes we take the frequency content of the signal as a triangle. (If you Fourier transform the signal in Figure 4.2 you will not get such a nice triangle.)

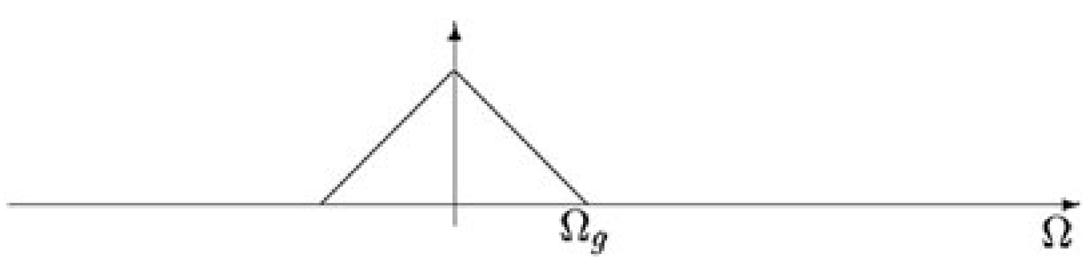


Figure 4.3: The spectrum $X(j\Omega)$.

Notice that the signal in Figure 4.3 is bandlimited. We can see that the signal is bandlimited because

$X(j\Omega)$ is zero outside the interval $[-\Omega_g, \Omega_g]$. Equivalentely we can state that the signal has no angular frequencies above Ω_g, corresponding to no frequencies above $F_g = \frac{\Omega_g}{2\pi}$.

Now let's take a look at the sampled signal in the frequency domain. While proving (Section 4.2) the sampling theorem we found the the spectrum of the sampled signal consists of a sum of shifted versions of the analog spectrum. Mathematically this is described by the following equation:

$$X_s\left(e^{j\Omega T_s}\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X\left(j\left(\Omega + \frac{2\pi k}{T_s}\right)\right) \tag{4.15}$$

4.3.2.1 Sampling fast enough

In Figure 4.4 we show the result of sampling $x(t)$ according to the sampling theorem (Section 4.1.4: The Sampling Theorem). This means that when sampling the signal in Figure 4.2/Figure 4.3 we use $F_s \geq 2F_g$. Observe in Figure 4.4 that we have the same spectrum as in Figure 4.3 for $\Omega \in [-\Omega_g, \Omega_g]$, except for the scaling factor $\frac{1}{T_s}$. This is a consequence of the sampling frequency. As mentioned in the proof (Key points in the proof, p. 47) the spectrum of the sampled signal is periodic with period $2\pi F_s = \frac{2\pi}{T_s}$.

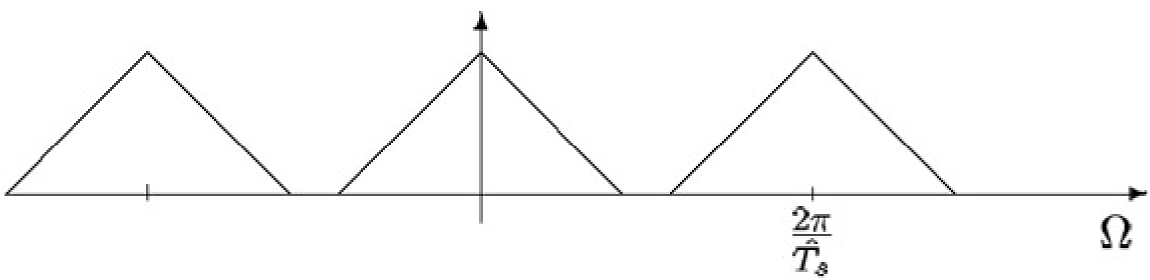


Figure 4.4: The spectrum X_s. Sampling frequency is OK.

So now we are, according to the sample theorem (Section 4.1.4: The Sampling Theorem), able to reconstruct the original signal **exactly**. How we can do this will be explored further down under reconstruction (Section 4.3.3: Reconstruction). But first we will take a look at what happens when we sample too slowly.

4.3.2.2 Sampling too slowly

If we sample $x(t)$ too slowly, that is $F_s < 2F_g$, we will get overlap between the repeated spectra, see Figure 4.5. According to (4.15) the resulting spectra is the sum of these. This overlap gives rise to the concept of aliasing.

> NOTE: If the sampling frequency is less than twice the highest frequency component, then frequencies in the original signal that are above half the sampling rate will be "aliased" and will appear in the resulting signal as lower frequencies.

The consequence of aliasing is that we cannot recover the original signal, so aliasing has to be avoided. Sampling too slowly will produce a sequence $x_s(n)$ that could have orginated from a number of signals. So there is **no** chance of recovering the original signal. To learn more about aliasing, take a look at this module (Section 4.5). (Includes an applet for demonstration!)

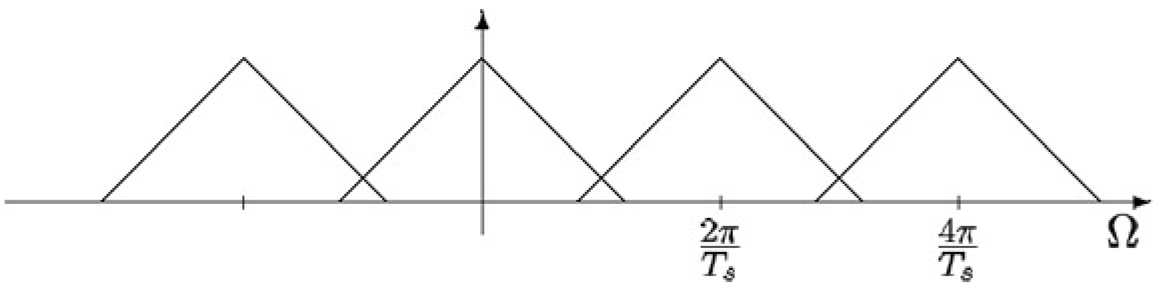


Figure 4.5: The spectrum X_s. Sampling frequency is too low.

To avoid aliasing we have to sample fast enough. But if we can't sample fast enough (possibly due to costs) we can include an Anti-Aliasing filter. This will not able us to get an exact reconstruction but can still be a good solution.

NOTE: Typically a low-pass filter that is applied before sampling to ensure that no components with frequencies greater than half the sample frequency remain.

Example 4.3
The stagecoach effect
In older western movies you can observe aliasing on a stagecoach when it starts to roll. At first the spokes appear to turn forward, but as the stagecoach increase its speed the spokes appear to turn backward. This comes from the fact that the sampling rate, here the number of frames per second, is too low. We can view each frame as a sample of an image that is changing continuously in time. (Applet illustrating the stagecoach effect[8])

4.3.3 Reconstruction

Given the signal in Figure 4.4 we want to recover the original signal, but the question is how?

When there is no overlapping in the spectrum, the spectral component given by $k = 0$ (see (4.15)),is equal to the spectrum of the analog signal. This offers an oppurtunity to use a simple reconstruction process. Remember what you have learned about filtering. What we want is to change signal in Figure 4.4 into that of Figure 4.3. To achieve this we have to remove all the extra components generated in the sampling process. To remove the extra components we apply an ideal analog low-pass filter as shown in Figure 4.6 As we see the ideal filter is rectangular in the frequency domain. A rectangle in the frequency domain corresponds to a sinc[9] function in time domain (and vice versa).

[8] http://flowers.ofthenight.org/wagonWheel/wagonWheel.html
[9] http://ccrma-www.stanford.edu/~jos/Interpolation/sinc_function.html

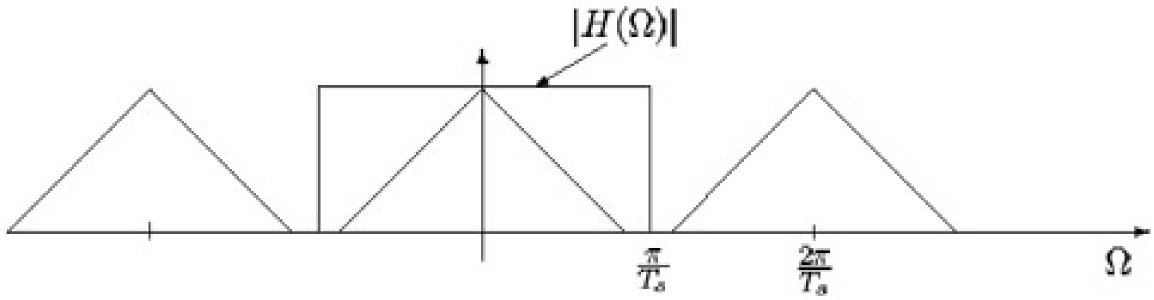


Figure 4.6: $H(j\Omega)$ The ideal reconstruction filter.

Then we have reconstructed the original spectrum, and as we know **if two signals are identical in the frequency domain, they are also identical in the time domain**. End of reconstruction.

4.3.4 Conclusions

The Shannon sampling theorem requires that the input signal prior to sampling is band-limited to at most half the sampling frequency. Under this condition the samples give an exact signal representation. It is truly remarkable that such a broad and useful class signals can be represented that easily!

We also looked into the problem of reconstructing the signals form its samples. Again the simplicity of the **principle** is striking: linear filtering by an ideal low-pass filter will do the job. However, the ideal filter is impossible to create, but that is another story...

4.3.5

Go to? Introduction (Section 4.1); Proof (Section 4.2); Illustrations (Section 4.3); Matlab Example (Section 4.4); Aliasing applet (Section 4.5); Hold operation (Section 4.6); System view (Section 4.7); Exercises (Section 4.8)

4.4 Sampling and reconstruction with Matlab[10]

4.4.1 Matlab files

Samprecon.m[11]

4.4.2

Introduction (Section 4.1); Proof (Section 4.2); Illustrations (Section 4.3); Aliasing applet (Section 4.5); Hold operation (Section 4.4); System view (Section 4.7); Exercises (Section 4.8)

4.5 Aliasing Applet[12]

The applet is courtesy of the Digital Signal Processing tutorial at freeuk.com, *http://www.dsptutor.freeuk.com/*. You can also have a look at the Light Wheel applet[13] .

[10] This content is available online at <http://cnx.org/content/m11549/1.9/>.
[11] http://cnx.rice.edu/content/m11549/latest/Samprecon.m
[12] This content is available online at <http://cnx.org/content/m11448/1.14/>.
[13] http://flowers.ofthenight.org/wagonWheel/lightWheel.html

4.5.1 Introduction

In this module we shall look at sampling a sinusoidal signal. According to the sampling theorem (Section 4.1.4: The Sampling Theorem), a sinusoidal signal can be exactly reconstructed from values sampled at discrete, uniform intervals as long as the signal frequency is less than half the sampling frequency. Any component of a sampled signal with a frequency above this limit, often referred to as the folding frequency, is subject to aliasing (p. 51).

The applet is based on a fixed sampling rate of $F_s = 8000 samplespersecond$ (one sample every 0.125 milliseconds, i.e $T_s = \frac{1}{8000}$).

4.5.2 Instructions

Set the frequency of the sinusoidal signal, in Hz, in the "Input frequency" box, i.e choose an f in the following signal: $\sin(2\pi f t)$. When you click the "Plot" button, with "Input signal" checked, the input signal is plotted against time.

The "Grid" checkbox toggles on and off vertical gridlines indicating the instants at which the signal is sampled. The "Sample points", representing the sampled values of the input signal, can also be toggled.

Finally, the "Alias frequency" checkbox (visible only when aliasing (p. 51) occurs) controls the plotting of the "reconstructed" sinusoidal signal, with $f = f_{\text{alias}}$.

4.5.3 Overview of the process

When using the applet it is important to have an understanding of where the different signals occur in a sampling system.

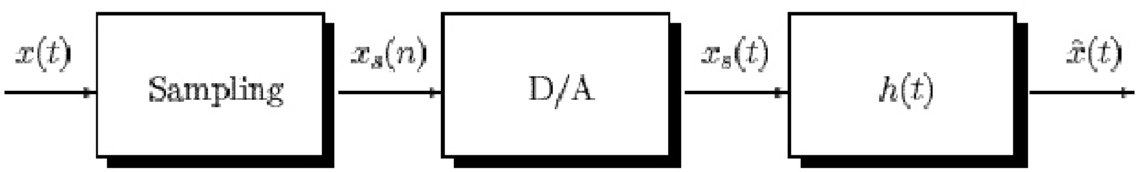


Figure 4.7: Ideal sampling process

Relating the applet signals to the figure we get

- Input signal $= x(t) = \sin(2\pi f t)$, where f is the input frequency chosen by the user.
- The sampled signal $= x_s(n) = \sin(2\pi f n T_s) = \sin\left(2\pi f n \times \frac{1}{8000}\right)$.
- The reconstructed signal $= \hat{x}((\text{t}))$, is shown as the original signal if sampling is done fast enough, or as the aliased signal if sampling is too slow.

($h(t)$ is an ideal reconstruction filter).

4.5.4 Aliasing demo applet

This is a Java Applet. To view, please see http://cnx.org/content/m11448/latest/

4.6 Hold operation[14]

Any practical reconstruction system must input finite length pulses into the reconstruction filter. The reason is that we need nonzero energy (Section 1.6.1: Signal Energy) in the nonzero pulses.

[14]This content is available online at <http://cnx.org/content/m11458/1.10/>.

4.6.1 Introduction

The operation performed to produce these pulses is called **hold**. Using the hold-operation we get pulses with a predefined length and height proportional to the input to the digital-to-analog converter. By means of the hold operation we get nonzero pulses with energy (Section 1.6.1: Signal Energy).

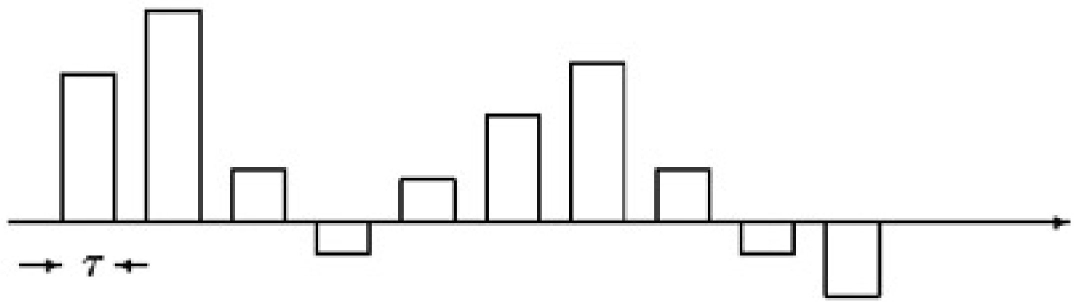


Figure 4.8: Output signal from the hold device

As we have made changes relative to the ideal reconstruction (p. 52), we need to look at the output signal the reconstruction filter will give us. Quite obviously the output will not be the original signal. So, is it still useful?

4.6.2 Analysis

As before, and as will be the situation later, using the frequency domain simplifies the analysis. To model the hold operation we use convolution (Chapter 8) with a delta function (Chapter 8) and a square pulse. The square pulse has unit height and duration τ. The duration τ is the **holding time**, i.e. how long we **hold** the incoming value. For the pulses not to overlap we must choose $\tau < T_s$. The convolution can be seen as a filtering operation, using the square pulse as the impulse response. If we fourier transform (Chapter 8) the square pulse we obtain the frequency response of the filter, which is a sinc[15] function.

Figure 4.9 shows the frequency response of the analog square pulse filter. We have plotted the frequency response for $\tau = T_s$ and $\tau = \frac{T_s}{2}$.

[15] http://ccrma-www.stanford.edu/~jos/Interpolation/sinc_function.html

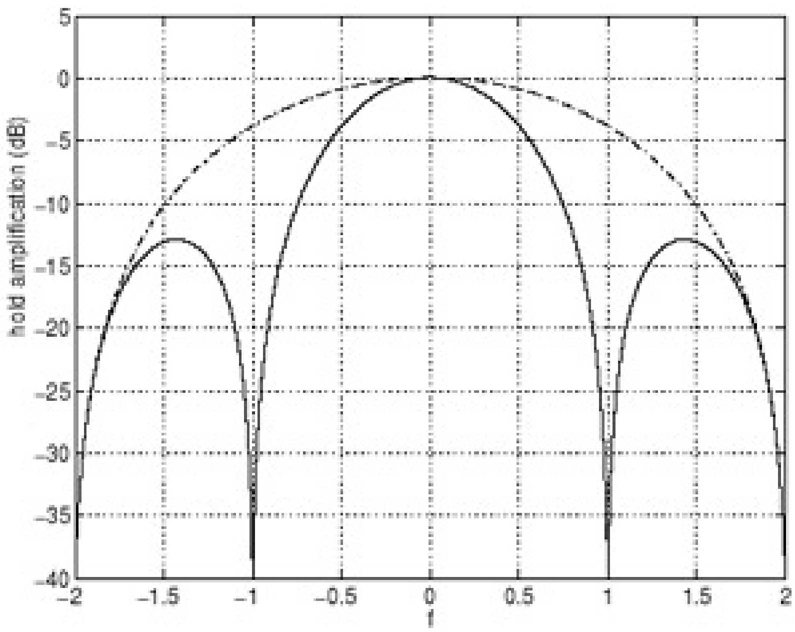


Figure 4.9: Frequency response of the analog square filter as a function of **digital frequency** f.

From the figure we can make the following observations

- The signal will be attenuated more and more towards the band edge, $f = 0.5$
- For $\tau = T_s$ the maximum attenuation is 3 dB at $f = 0.5$.
- For $\tau = \frac{T_s}{2}$ the maximum attenuation is 0.82 dB at $f = 0.5$.

The distortion is a result of linear operations and can thus be compensated for by using a filter with opposite frequency response in the passband, $f \in [-0.5, 0.5]$. The compensation will not be exact, but we can make the approximation as accurate as we wish. The compensation can be made in the reconstruction filter or after the reconstruction by using a separate analog filter. One can also predistort the signal in a digital filter before reconstruction. Where to put the compensator and it's quality are cost considerations.

4.6.3

Go to? Introduction (Section 4.1); Proof (Section 4.2); Illustrations (Section 4.3); Aliasing applet (Section 4.5); System view (Section 4.7); Exercises (Section 4.8)

4.7 Systems view of sampling and reconstruction[16]

4.7.1 Ideal reconstruction system

Figure 4.10 shows the ideal reconstruction system based on the results of the Sampling theorem proof (Section 4.2).

Figure 4.10 consists of a sampling device which produces a time-discrete sequence $x_s(n)$. The reconstruction filter, $h(t)$, is an ideal analog sinc[17] filter, with $h(t) = \text{sinc}\left(\frac{t}{T_s}\right)$. We can't apply the time-discrete

[16] This content is available online at <http://cnx.org/content/m11465/1.20/>.
[17] http://ccrma-www.stanford.edu/~jos/Interpolation/sinc_function.html

sequence $x_s(n)$ directly to the analog filter $h(t)$. To solve this problem we turn the sequence into an analog signal using delta functions (Chapter 8). Thus we write $x_s(t) = \sum_{n=-\infty}^{\infty} x_s(n)\,\delta(t - nT)$.

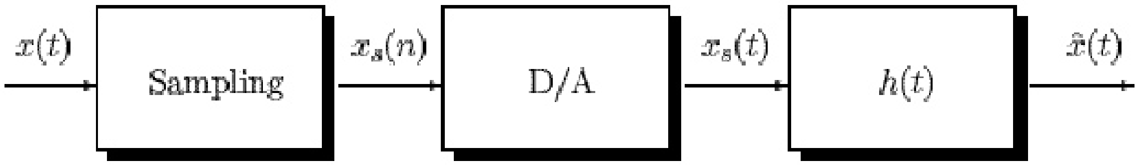


Figure 4.10: Ideal reconstruction system

But when will the system produce an output $\hat{x}(t) = x(t)$? According to the sampling theorem (Section 4.1.4: The Sampling Theorem) we have $\hat{x}(t) = x(t)$ when the sampling frequency, F_s, is at least twice the highest frequency component of $x(t)$.

4.7.2 Ideal system including anti-aliasing

To be sure that the reconstructed signal is free of aliasing it is customary to apply a lowpass filter, an anti-aliasing filter (p. 52), before sampling as shown in Figure 4.11.

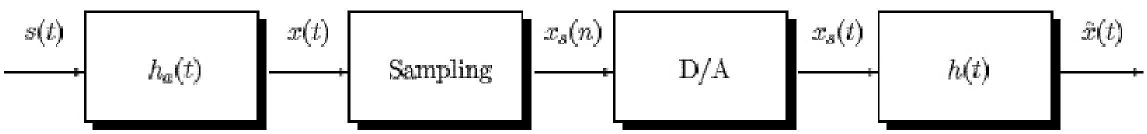


Figure 4.11: Ideal reconstruction system with anti-aliasing filter (p. 52)

Again we ask the question of when the system will produce an output $\hat{x}(t) = s(t)$? If the signal is entirely confined within the passband of the lowpass filter we will get perfect reconstruction if F_s is high enough.

But if the anti-aliasing filter removes the "higher" frequencies, (which in fact is the job of the anti-aliasing filter), we will **never** be able to **exactly** reconstruct the original signal, $s(t)$. If we sample fast enough we can reconstruct $x(t)$, which in most cases is satisfying.

The reconstructed signal, $\hat{x}(t)$, will not have aliased frequencies. This is essential for further use of the signal.

4.7.3 Reconstruction with hold operation

To make our reconstruction system realizable there are many things to look into. Among them are the fact that any practical reconstruction system must input finite length pulses into the reconstruction filter. This can be accomplished by the hold operation (Section 4.6). To alleviate the distortion caused by the hold opeator we apply the output from the hold device to a compensator. The compensation can be as accurate as we wish, this is cost and application consideration.

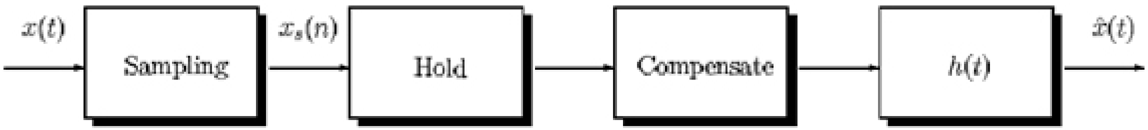


Figure 4.12: More practical reconstruction system with a hold component (Section 4.6)

By the use of the hold component the reconstruction will not be exact, but as mentioned above we can get as close as we want.

4.7.4

Introduction (Section 4.1); Proof (Section 4.2); Illustrations (Section 4.3); Matlab example (Section 4.4); Hold operation (Section 4.6); Aliasing applet (Section 4.5); Exercises (Section 4.8)

4.8 Exercises[18]

Problems related to the Sampling Theorem module. (Section 4.1)

Exercise 4.8.1 *(Solution on p. 60.)*
Express the sampling theorem in words.

Exercise 4.8.2 *(Solution on p. 60.)*
Theoretically, why is the sinc-function so important for reconstruction? Sketch a sinc(t). What are the values for integer values of t?

Exercise 4.8.3 *(Solution on p. 60.)*
Argue that the sampling rate for CD should be over 40KHz.

Exercise 4.8.4 *(Solution on p. 60.)*
(By Don Johnson)
What is the simplest bandlimited signal? Using this signal, convince yourself that less than two samples/period will not suffice to specify it. If the sampling rate $\frac{1}{T_s}$ is not high enough, what signal would your resulting undersampled signal become? Hint: Try the aliasing applet (Section 4.5).

Exercise 4.8.5 *(Solution on p. 60.)*
Are the filter h(t) described by the sinc function the only filter we can use as a perfect reconstruction filter? If not what are the condition that would allow us to use another filter?

Exercise 4.8.6 *(Solution on p. 60.)*
If you found that it is possible to use another filter in Exercise 4.8.5 specify such a filter. Hint: Try using the domain which usually simplifies things...

Exercise 4.8.7 *(Solution on p. 60.)*
What are the difficulties introduced when we want to apply the results of this chapter in practice?

Exercise 4.8.8 *(Solution on p. 60.)*
If a real signal has frequency content up to F_1. What is then the bandwith of the signal?

Exercise 4.8.9 *(Solution on p. 60.)*
If a real signal has frequency content confined in the interval $[-F_1, F_1]$. What is then the bandwith of the signal?

[18]This content is available online at <http://cnx.org/content/m11442/1.16/>.

Exercise 4.8.10 *(Solution on p. 60.)*
What can be said in general for the spectrum of a discrete signal which is the result of sampling an analog signal that is NOT bandlimited?

4.8.1 Exercises related to the Aliasing applet

Link to the aliasing applet (Section 4.5) (Right click if you want to open it in a new window).

In the following problems, as in the aliasing applet, we are studying a sinusoidal signal, $x(t) = \sin(2\pi f t)$, which is sampled at $F_s = 8000$.

Exercise 4.8.11 *(Solution on p. 60.)*
What is the frequency limitation of an analog sinusoidal signal if we want to avoid aliasing, given $F_s = 8000$?

Exercise 4.8.12 *(Solution on p. 60.)*
Describe with words the type of signal we "reconstruct" from the samples when the input frequency (of the sinusoidal signal) is higher than the sample rate can deal with?

Exercise 4.8.13 *(Solution on p. 60.)*
Find an expression the signal we "reconstruct" from the samples when the input frequency is 6000 Hz.

Exercise 4.8.14 *(Solution on p. 60.)*
Explain the "strange" sample points when the input input frequency is 4000 Hz.

Exercise 4.8.15 *(Solution on p. 60.)*
Explain the "strange" sample points when the input input frequency is 8000 Hz.

Exercise 4.8.16 *(Solution on p. 60.)*
Find an expression for the signal we can reconstruct from the samples when the input frequency is 4000 Hz.

Exercise 4.8.17 *(Solution on p. 61.)*
Find an expression for the "reconstructed" signal from the samples when the input frequency is 8000 Hz.

Solutions to Exercises in Chapter 4

Solution to Exercise 4.8.1 (p. 58)
Fill in the solution here...
Solution to Exercise 4.8.2 (p. 58)
Fill in the solution here...
Solution to Exercise 4.8.3 (p. 58)
The human ear can hear frequencies up to 20 KHz, so according to the sampling theorem we should sample at a rate equal to or exceeding 40KHz. In practice we always have to sample at more than the double rate, partly due to finite precision.
Solution to Exercise 4.8.4 (p. 58)
The simplest bandlimited signal is the sine wave. At the Nyquist frequency, exactly two samples/period would occur. Reducing the sampling rate would result in fewer samples/period, and these samples would appear to have arisen from a lower frequency sinusoid.
Solution to Exercise 4.8.5 (p. 58)
Fill in a solution here
Solution to Exercise 4.8.6 (p. 58)
Fill in a solution here
Solution to Exercise 4.8.7 (p. 58)
Fill in a solution here
Solution to Exercise 4.8.8 (p. 58)
Fill in a solution here
Solution to Exercise 4.8.9 (p. 58)
Fill in a solution here
Solution to Exercise 4.8.10 (p. 59)
The spectrum will ALWAYS overlap,there will always be aliasing.
Solution to Exercise 4.8.11 (p. 59)
With a sampling frequency of 8000 Hz, the maximum frequency of the analog signal is 4000 Hz, as given by the sampling theorem (Section 4.1.4: The Sampling Theorem).
Solution to Exercise 4.8.12 (p. 59)
The signal we "reconstruct" is a sinusoidal signal with a frequency that is lower than the original because of aliasing.
Solution to Exercise 4.8.13 (p. 59)
When the input frequency is 6000 Hz, a sampling frequency of 8000 Hz is to low, i.e aliasing will occur. The sampled signal will have frequency components at +6000 Hz and -6000 Hz plus some new frequency components as a result of aliasing.

We know from the proof of the sampling theorem (Section 4.2.1: Introduction) that the sampled signal is periodic with $F_s = 8000$. Thus a frequency component at 6000 Hz implies frequencies at -2000 Hz, -10000 Hz, 14000 Hz and so on. Similarly a frequency component at -6000 Hz give rise to(among others) a 2000 Hz component. Looking only at the positive frequencies the "reconstructed" signal will only have a 2000 Hz frequency component. The removal of the 6000 Hz and above frequencies are due to the reconstruction filter. The filter is designed based on a maximum input signal frequency of 4000 Hz. Thus the "reconstructed" signal can be written as: $\sin(2\pi 2000t)$.
Solution to Exercise 4.8.14 (p. 59)
The sampled signal can be written as $x_s(n) = \sin\left(2\pi 4000\frac{n}{8000}\right) = \sin(\pi n) = 0$. Thus all the samples are zero-valued.
Solution to Exercise 4.8.15 (p. 59)
The sampled signal can be written as $x_s(n) = \sin\left(2\pi 8000\frac{n}{8000}\right) = \sin(2\pi n) = 0$. Thus all the samples are zero-valued.
Solution to Exercise 4.8.16 (p. 59)
As shown in problem 14, the samples are zero valued. A reconstructing filter cannot distinguish this from the all zero signal so the reconstructed signal will be the all zero signal.

Note that a small change in the sinusoidal signals phase would produce samples that are not only zero-valued. The "reconstructed" signal will then be a equal to the original signal. This problem illustrates that sampling twice the signals highest frequency component does not always guarantee perfect recontstruction. If we could increase the sampling frequency to, say, $F_s = 8000.00001$, we could reconstruct the original signal. I.e sampling at a rate **greater** than twice the highest frequency component yields the desired reconstruction.

Solution to Exercise 4.8.17 (p. 59)

As shown in problem 15, the samples are zero valued. A reconstructing filter cannot distinguish this from the all zero signal so the reconstructed signal will be the all zero signal.

Note that a small change in the sinusoidal signals phase would produce samples that are not only zero-valued. The "reconstructed" signal will then be a signal with aliased components.

Chapter 5

Information theory

5.1 Introduction[1]

In this and the following modules the basic concepts of information theory will be introduced. For simplicity we assume that the signals are time discrete. Time discrete signals often arise from sampling a time continous signal. The assumption of time discrete signal is valid because we will only be looking at bandlimited signals (Figure 4.3). (Which can, as we know (Section 4.1.4: The Sampling Theorem), be perfectly reconstructed).

In treating time discrete signal and their information content we have to distinguish between two types of signals:

- signals have amplitude levels belonging to a **finite** set
- signals that have amplitudes taken from the real line

In the first case we can measure the information content in terms of entropy (Section 5.4), while in the second case the entropy is infinte and we must resort to characterise the source by means of differential entropy (Section 5.5).

5.1.1 Examples of information sources

The signals treated are mainly of a stochastic nature, i.e. the signal is unknown to us. Since the signal is not known to the destination (because of it's stochastic nature), it is then best modeled as a random process, discrete-time or continuous time. Examples of information sources that we model as random processes are:

- Digital data source (e.g. a text) can be modeled as a random process.
- Video signals can be modeled as a random process. Such signals are mainly bandlimited to around 5 MHz (the value depends on the standards used to raster the frames of image).
- Audio signals can be modeled as a random process. Speech is typically between 300 Hz and 3400 Hz, see Figure 5.1.

[1]This content is available online at <http://cnx.org/content/m11838/1.5/>.

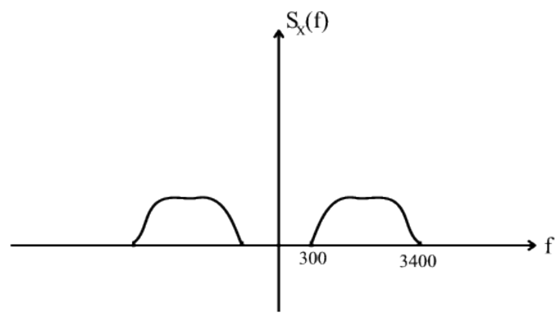


Figure 5.1: Power spectral density plot of speech

Video and speech are analog information signals are bandlimited. Therefore, if sampled faster than two times the highest fequency component, they can be reconstructed from their sample values.

Example 5.1
A speech signal with bandwidth of 3100 Hz can be sampled at the rate of 6.2 KHz. If the samples are quantized with a 8 level quantizer then the speech signal can be represented with a binary sequence with bit rate

$$6200 \log_2 8 = 18600 \text{bits/sec} \tag{5.1}$$

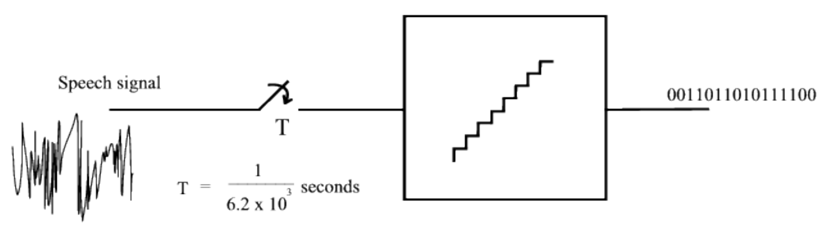


Figure 5.2: Analog speech signal sampled and quantised

The sampled real values can be quantized to create a discrete-time discrete-valued random process.

5.1.2 The Core of Information theory

The key observation from the discussion above is that for a reveiver the signals are **unknown**. It is exact this uncertainty that enables the signal to transmit information. This is the core of information theory:

NOTE: Information transfer happens when the receiver is unable to know or predict at message before it is received.

5.1.3 Some statistics

Here we present some statistics with the intent of reviewing a few basic concepts and to introduce the notation.

Let X be a **stochastic** variable. Let $X = x_i$ and $X = x_j$ denote two outcomes of X.

- **Dependent** outcomes implies: $Pr[X = x_i, X = x_j] = Pr[X = x_i] Pr[X = x_j \mid x_i] = Pr[X = x_j] Pr[X = x_i \mid x_j]$
- **Independent** outcomes implies $Pr[X = x_i, X = x_j] = Pr[X = x_i] Pr[X = x_j]$
- Bayes' rule: $Pr[X = x_j \mid x_i] = \frac{Pr[X=x_i \mid x_j] Pr[X=x_j]}{Pr[X=x_i]}$

More about basic probability theory and a derivation of Bayes' rule can be found here[2].

5.2 Information[3]

In this module we introduce the concept of self information for an outcome of a stochastic variable.

5.2.1

Example 5.2

Bergen, Norway is a rainy city. If the locals are "lucky" there is "only" 200 rainy days in a particular year. Let the random variable Z take the two values: "Rain", "No rain". Assuming 200 rainy days a year, we get $Pr[Z = \text{Rain}] = \frac{200}{365}$ and $Pr[Z = \text{No Rain}] = \frac{165}{365}$. We state that $Z = \text{No Rain}$ carries more information than $Z = \text{Rain}$, the reason is that the inhabitans of Bergen expect rain, so whenever it's not raining they are (more) surprised. An intuitive definition of an information measure should be larger when the probability is small.

Example 5.3

The information content in a statement about the temperature and new lottery millionaires in Verdal,Norway on a given saturday should be **the sum** of the information on temperature on the particular saturday in Verdal and the information of the number of new lucky lottery winners, (under the assumption that these observations are independent). Let I denote the information of an event, then

$$I(\text{temperature}, \text{lottery winners}) = I(\text{temperature}) + I(\text{lottery winners}) \quad (5.2)$$

5.2.2 The self information formula

An intuitive and meaningful measure of self information in an event should have the following properties:

1. The more **uncertain** you, in advance, are about the outcome, the **more new information** you get by observing the actual outcome, or equivalently an event with low probability, p_n, has high self information $I(p_n)$. $I(p_n)$ should be a monotonically decreasing function of p_n.
2. Oberserving an event with certain outcome, i.e $p_n = 1$, should give zero information. The event p_n is then said to have zero self information. Since $I(p_n)$ is monotonically decreasing for $p_n \in [0, 1]$ this implies that the self information can never be less than zero, the observer can never lose information by observing an outcome.

[2] "Foundations of Probability Theory: Basic Definitions" <http://cnx.org/content/m11245/latest/>

[3] This content is available online at <http://cnx.org/content/m11841/1.2/>.

3. If we receive independent messages, the information should accumulate. This means that the measure must be additive.

It can be shown that there only exists one function satisfying the above conditions.

NOTE: $I(p_n) = \log_b \frac{1}{p_n} = -\log_b p_n$

In the above equation the logarithm base can be chosen arbitrary. Usually $b = 2$ is chosen so that the denomination is **information bit**. The choice $b = 2$ is made to adapt to a digital "world", that is to facilitate electronic storage and transmission.

5.3 Representing symbols by bits[4]

5.3.1 Introduction

Often we want to represent data, e.g. characters, images, in a binary form. By binary form we mean representing by the symbols "0", and "1". Using binary representation allows us to conveniently store, retrieve, and manipulate them with a computer. To work with data in binary form we must have a fixed way of encoding (representing) a fixed data stream. The set of all binary sequences in a representation of some data is called a **code**. (Note that this has nothing to do with cryptology). Usually we refer to the data that we want to represent by bits as a **source**.

Example 5.4: Representing English Characters

Let us consider a very practical example of the above ideas. Let our source be a stream of English characters. Now we want to represent this stream of characters as bits, say to store it on a computer or send it over the Internet. First we need to know the number of such characters, which is (traditonally) conveniently set to 128. The number 128 is obtained by summing upper case charachters (26), lower case (26), digits (10), brackets and punctuation (20), odd characters (14) (the "&" is an odd character), and control characters (32).

Obviously we need to have a unique representation of each of the 128 characters, this can e.g. be obtained by exhausting the 128 bit combinations which concatenating 7 bits give. Thus we have devised an 7-bit code. A well known 7-bit code is **ASCII**, short for "American Standard Code for Information Interchange". Adding a parity bit for error control to the ASCII code forms an 8-bit code. As an example, the representation of an "A" in ASCII is 1000001.

Now, one can ask whether the 7-bit ASCII code is an optimal representation in terms of using, on average, the minimum number of bits representing the English characters? We will return to this question later (in example 3 (Example 5.6: Optimality of the ASCII code)).

5.3.2 Minimal representation

When representing a source we want to use as few bits as possible, as this will imply that less disk space is required for storage or that transmission over the Internet is quicker. However, we do not want to use so few bits that the receiver cannot determine what was sent or stored.

So, for a given source what is the minimal representation? Here we consider the minimal representation as the representation that uses the minimum number of bits (on average) to encode the source without errors. According to Shannon's[5] **source coding theorem**, a source that produces statistically independent outcomes, the minimum average number of bits per symbol is the entropy (Section 5.4) of the source! (A classical example of a source that produces statistically independent outcomes is throwing a die.)

Average indicates that the number of bits used for a specific symbol may be different from the number of bits representing another. E.g., as opposed to ASCII coding, we might represent an "A" with 7 bits, but

[4] This content is available online at <http://cnx.org/content/m11869/1.2/>.
[5] http://en.wikipedia.org/wiki/Claude_Shannon

an "E" with 3 bits. But it also implies that when you receive a series of symbols, the number you receive per time unit, say per second, will not be exactly the same, but averaged over a long term period, the rate is proportional to time with the rate per symbol as the proportionality constant.

Let us assume that we represent a symbol x_n, with probability p_n, by l_n bits. Then, the average number of bits spent per symbol will be

$$\bar{L}= \sum_{n=1}^{N} p_n l_n \tag{5.3}$$

We see that this equation is equal to the entropy if the code words are selected to have the lengths $l_n = -\log p_n$. Thus, if the source produces stochastically independent outcomes with probabilities p_n, such that $\log p_n$ is an integer, then we can easily find an optimal code as we show in the next example.

Example 5.5: Finding a minimal representation

A four-symbol alphabet produces stochastically independent outcomes with the following probabilities.

$$Pr\left[x_1\right] = \frac{1}{2}$$

$$Pr\left[x_2\right] = \frac{1}{4}$$

$$Pr\left[x_3\right] = \frac{1}{8}$$

$$Pr\left[x_4\right] = \frac{1}{8}$$

and an entropy of 1.75 bits/symbol. Let's see if we can find a codebook for this four-letter alphabet that satisfies the Source Coding Theorem. The simplest code to try is known as the **simple binary code**: convert the symbol's index into a binary number and use the same number of bits for each symbol by including leading zeros where necessary.

$$x_1 \leftrightarrow 00 \; x_2 \leftrightarrow 01 \; x_3 \leftrightarrow 10 \; x_4 \leftrightarrow 11 \tag{5.4}$$

As all symbols are represented by 2 bits, obviously the average number of bits per symbol is 2. Because the entropy equals 1.75 bits, the simple binary code is not a minimal representation according to the source coding theorem. If we chose a codebook with differing number of bits for the symbols, a smaller average number of bits can indeed be obtained. The idea is to **use shorter bit sequences for the symbols that occur more often**, i.e., symbols that have a higher probability. One codebook like this is

$$x_1 \leftrightarrow 0 \; x_2 \leftrightarrow 10 \; x_3 \leftrightarrow 110 \; x_4 \leftrightarrow 111 \tag{5.5}$$

Now $\bar{L}= 1 \times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{1}{8} + 3 \times \frac{1}{8} = 1.75$. We can reach the entropy limit! This should come as no surprise, as promised above, when $\log p_n$ is an integer for all n, the optimal code is easily found.

The simple binary code is, in this case, less efficient than the unequal-length code. Using the efficient code, we can transmit the symbolic-valued signal having this alphabet 12.5% faster. Furthermore, we know that no more efficient codebook can be found because of Shannon's source coding theorem.

Example 5.6: Optimality of the ASCII code

Let us return to the ASCII codes presented in Example 5.4 (Representing English Characters). Is the 7-bit ASCII code optimal, i.e., is it a minimal representation? The 7-bit ASCII code assign an equal length (7-bit) to all characters it represents. Thus, it would be optimal if all of the 128 characters were equiprobable, that is each character should have a probability of $\frac{1}{128}$. To find out whether the characters really are equiprobable an analysis of all English texts would be needed.

Such an analysis is difficult to do. However, the letter "E" is more probable than the letter "Z", so the equiprobable assumption does not hold, and the ASCII code is not optimal.

(A technical note: We should take into account that in English text subsequent outcomes are not stochastically independent. To see this, assume the first letter to be "b", then it is more probable that the next letter is "e", than "z". In the case where the outcomes are not stochastically independent, the formulation we have given of Shannon's source coding theorem is no longer valid, to fix this, we should replace the entropy with the entropy rate, but we will not pursue this here).

5.3.3 Generating efficient codes

From Shannon's source coding theorem we know what the minimum average rate needed to represent a source is. But other than in the case when the logarithm of the probabilities gives an integer, we do not get any indications on how to obtain that rate. It is a large area of research to get close to the Shannon entropy bound. One clever way to do encoding is the Huffman coding (Section 5.6) scheme.

5.4 Entropy[6]

5.4.1

The self information (Section 5.2) gives the information in a single outcome. In most cases, e.g in data compression, it is much more interesting to know the **average information content** of a source. This average is given by the **expected** value of the self information with respect to the source's probability distribution. This average of self information is called the source entropy.

5.4.1.1 Definition of entropy

Definition 5.1: Entropy
1. The entropy (average self information) of a discrete random variable X is a function of its probability mass function and is defined as

$$H(X) = -\sum_{i=1}^{N} p_X(x_i) \log p_X(x_i) \tag{5.6}$$

where N is the number of possible values of X and $P_X(x_i) = Pr[X = x_i]$. If log is base 2 then the unit of entropy is bits per (source)symbol. Entropy is a measure of uncertainty in a random variable and a measure of information it can reveal.
2. If symbol has zero probability, which means it never occurs, it should not affect the entropy. Letting $0 \times \log 0 = 0$, we have dealt with that.

In texts you will find that the argument to the entropy function may vary. The two most common are $H(X)$ and $H(p)$. We calculate the entropy of a source X, but the entropy is, strictly speaking, a function of the source's probabilty function p. So both notations are justified.

5.4.1.2 Calculating the binary logarithm

Most calculators does not allow you to directly calculate the logarithm with base 2, so we have to use a logarithm base that most calculators support. Fortunately it is easy to convert between different bases.

Assume you want to calculate $\log_2 x$, where $x > 0$. Then $\log_2 x = y$ implies that $2^y = x$. Taking the natural logarithm on both sides we obtain

NOTE: $\log_2 x = \frac{\ln(x)}{\ln(2)}$

[6] This content is available online at <http://cnx.org/content/m11839/1.4/>.

5.4.1.3 Examples

Example 5.7

When throwing a dice, one may ask for the average information conveyed in a single throw. Using the formula for entropy we get $H(X) = -\sum_{i=1}^{6} p_X(x_i) \log p_X(x_i) = \log 6 \text{bits/symbol}$

Example 5.8

If a soure produces binary information $\{0, 1\}$ with probabilities p and $1 - p$. The entropy of the source is

$$H(X) = (-(p\log_2 p)) - (1 - p)\log_2(1 - p) \tag{5.7}$$

If $p = 0$ then $H(X) = 0$, if $p = 1$ then $H(X) = 0$, if $p = 1/2$ then $H(X) = 1$. The source has its largest entropy if $p = 1/2$ and the source provides no new information if $p = 0$ or $p = 1$.

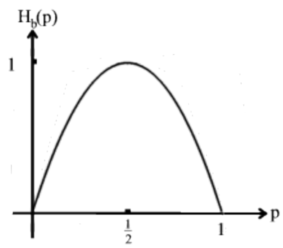


Figure 5.3

Example 5.9

An analog source is modeled as a continuous-time random process with power spectral density bandlimited to the band between 0 and 4000 Hz. The signal is sampled at the Nyquist rate. The sequence of random variables, as a result of sampling, are assumed to be independent. The samples are quantized to 5 levels $\{-2, -1, 0, 1, 2\}$. The probability of the samples taking the quantized values are $\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16}\right\}$, respectively. The entropy of the random variables are

$$\begin{aligned} H(X) &= -\sum_{i=1}^{5} p_X(x_i) \log p_X(x_i) \\ &= \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{1}{4} + \frac{1}{4} \\ &= \frac{15}{8} \text{bits/sample} \end{aligned} \tag{5.8}$$

There are 8000 samples per second. Therefore, the source produces $8000 \times \frac{15}{8} = 15000$ bits/sec of information.

Entropy is closely tied to source coding. The extent to which a source can be compressed is related to its entropy. There are many interpretations possible for the entropy of a random variable, including

- (Average)Self information in a random variable
- Minimum number of bits per **source symbol** required to describe the random variable without loss
- Description complexity
- Measure of uncertainty in a random variable

5.4.2 References

- Øien, G.E. and Lundheim,L. (2003) **Information Theory, Coding and Compression**, Trondheim: Tapir Akademisk forlag.

5.5 Differential Entropy[7]

Consider the entropy of **continuous** random variables. Whereas the (normal) entropy (Section 5.4) is the entropy of a **discrete** random variable, the differential entropy is the entropy of a continuous random variable.

5.5.1 Differential Entropy

Definition 5.2: Differential entropy
The differential entropy $h(X)$ of a continuous random variable X with a pdf $f(x)$ is defined as

$$h(X) = -\int_{-\infty}^{\infty} f(x) \log f(x) \, dx \quad (5.9)$$

Usually the logarithm is taken to be base 2, so that the unit of the differential entropy is bits/symbol. Note that is the discrete case, $h(X)$ depends only on the pdf of X. Finally, we note that the differential entropy is the expected value of $-\log f(x)$, i.e.,

$$h(X) = -E(\log f(x)) \quad (5.10)$$

Now, consider a calculating the differential entropy of some random variables.

Example 5.10
Consider a uniformly distributed random variable X from c to $c + \Delta$. Then its density is $\frac{1}{\Delta}$ from c to $c + \Delta$, and zero otherwise.

We can then find its differential entropy as follows,

$$\begin{aligned} h(X) &= -\int_c^{c+\Delta} \frac{1}{\Delta} \log \frac{1}{\Delta} dx \\ &= \log \Delta \end{aligned} \quad (5.11)$$

Note that by making Δ arbitrarily small, the differential entropy can be made arbitrarily negative, while taking Δ arbitrarily large, the differential entropy becomes arbitrarily positive.

Example 5.11
Consider a normal distributed random variable X, with mean m and variance σ^2. Then its density is $\sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{(x-m)^2}{2\sigma^2}}$.

We can then find its differential entropy as follows, first calculate $-\log f(x)$:

$$-\log f(x) = \frac{1}{2} \log \left(2\pi\sigma^2\right) + \log e \frac{(x-m)^2}{2\sigma^2} \quad (5.12)$$

Then since $E\left((X-m)^2\right) = \sigma^2$, we have

$$\begin{aligned} h(X) &= \frac{1}{2} \log \left(2\pi\sigma^2\right) + \frac{1}{2} \log e \\ &= \frac{1}{2} \log \left(2\pi e \sigma^2\right) \end{aligned} \quad (5.13)$$

[7] This content is available online at <http://cnx.org/content/m11840/1.3/>.

5.5.2 Properties of the differential entropy

In the section we list some properties of the differential entropy.

- The differential entropy can be negative
- $h(X+c) = h(X)$, that is translation **does not** change the differential entropy.
- $h(aX) = h(X) + \log|a|$, that is scaling **does** change the differential entropy.

The first property is seen from both Example 5.10 and Example 5.11. The two latter can be shown by using (5.9).

5.6 Huffman Coding[8]

One particular source coding[9] algorithm is the Huffman encoding algorithm. It is a source coding algorithm which approaches, and sometimes achieves, Shannon's bound for source compression. A brief discussion of the algorithm is also given in another module[10].

5.6.1 Huffman encoding algorithm

1. Sort source outputs in decreasing order of their probabilities
2. Merge the two least-probable outputs into a single output whose probability is the sum of the corresponding probabilities.
3. If the number of remaining outputs is more than 2, then go to step 1.
4. Arbitrarily assign 0 and 1 as codewords for the two remaining outputs.
5. If an output is the result of the merger of two outputs in a preceding step, append the current codeword with a 0 and a 1 to obtain the codeword the the preceding outputs and repeat step 5. If no output is preceded by another output in a preceding step, then stop.

Example 5.12
$X \in \{A, B, C, D\}$ with probabilities $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\}$

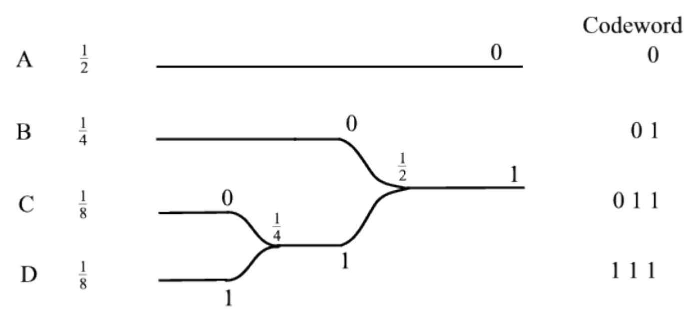


Figure 5.4

Average length $= \frac{1}{2}1 + \frac{1}{4}2 + \frac{1}{8}3 + \frac{1}{8}3 = \frac{14}{8}$. As you may recall, the entropy of the source was also $H(X) = \frac{14}{8}$. In this case, the Huffman code achieves the lower bound of $\frac{14}{8} \frac{\text{bits}}{\text{output}}$.

[8]This content is available online at <http://cnx.org/content/m10176/2.10/>.
[9]"Source Coding" <http://cnx.org/content/m10175/latest/>
[10]"Compression and the Huffman Code" <http://cnx.org/content/m0092/latest/>

In general, we can define average code length as

$$\bar{\ell} = \sum_{x \in \overline{X}} p_X(x)\,\ell(x) \tag{5.14}$$

where $\overline{X}$ is the set of possible values of x.

It is not very hard to show that

$$H(X) \geq \bar{\ell} > H(X) + 1 \tag{5.15}$$

For compressing single source output at a time, Huffman codes provide nearly optimum code lengths.

The drawbacks of Huffman coding

1. Codes are variable length.
2. The algorithm requires the knowledge of the probabilities, $p_X(x)$ for all $x \in \overline{X}$.

Another powerful source coder that does not have the above shortcomings is Lempel and Ziv.

Chapter 6

Decibel scale with signal processing applications[1]

6.1 Introduction

The concept of decibel originates from telephone engineers who were working with power loss in a telephone line consisting of cascaded circuits. The power loss in each circuit is the ratio of the power in to the power out, or equivivalently, the power gain is the ratio of the power out to the power in.

Let P_{in} be the power input to a telephone line and P_{out} the power out. The power gain is then given by

$$\text{Gain} = \frac{P_{\text{out}}}{P_{\text{in}}} \tag{6.1}$$

Taking the logarithm of the gain formula we obtain a comparative measure called Bel.

NOTE: $\text{Gain (Bel)} = \log\frac{P_{\text{out}}}{P_{\text{in}}}$

This measure is in honour of Alexander G. Bell, see Figure 6.1.

[1]This content is available online at <http://cnx.org/content/m12452/1.9/>.

Figure 6.1: Alexander G. Bell

6.2 Decibel

Bel is often a to large quantity, so we define a more useful measure, decibel:

$$\text{Gain}\,(\text{dB}) = 10\text{log}\frac{P_{\text{out}}}{P_{\text{in}}} \tag{6.2}$$

Please note from the definition that the gain in dB is relative to the input power. In general we define:

$$\text{Number of decibels} = 10\text{log}\frac{P}{P_{\text{ref}}} \tag{6.3}$$

If no reference level is given it is customary to use $P_{\text{ref}} = 1W$, in which case we have:

NOTE: Number of decibels $= 10\text{log}P$

Example 6.1
Given the power spectrum density (psd) function of a signal $x(n)$, $S_{\text{xx}}(jf)$. Express the magnitude of the psd in decibels.
We find $S_{\text{xx}}(\text{dB}) = 10\text{log}|S_{\text{xx}}(jf)|$.

6.3 More about decibels

Above we've calculated the decibel equivalent of power. Power is a quadratic variable, whereas voltage and current are linear variables. This can be seen, for example, from the formulas $P = \frac{V^2}{R}$ and $P = I^2R$.

So if we want to find the decibel value of a current or voltage, or more general an amplitude we use:

$$\text{Amplitude}\,(\text{dB}) = 20\log\frac{\text{Amplitude}}{\text{Amplitude}_{\text{ref}}} \tag{6.4}$$

This is illustrated in the following example.

Example 6.2

Express the magnitude of the filter $H\,(jf)$ in dB scale.

The magnitude is given by $|H\,(jf)\,|$, which gives: $|H\,(\text{dB})\,| = 20\log|H\,(jf)\,|$.

Plots of the magnitude of an example filter $|H\,(jf)\,|$ and its decibel equivalent are shown in Figure 6.2.

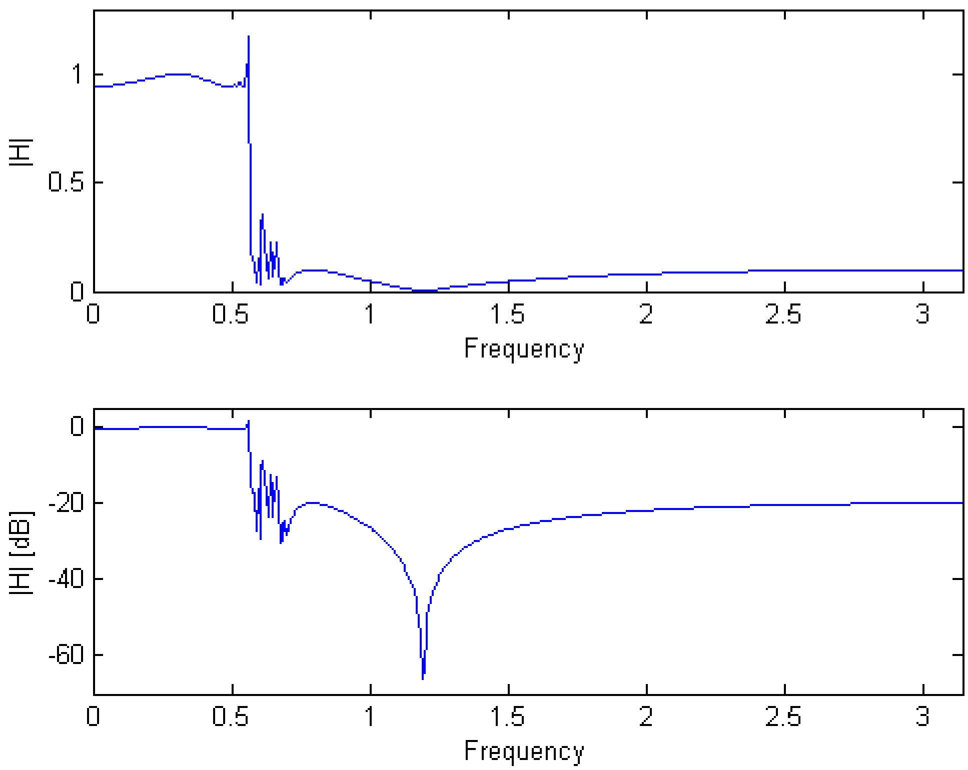


Figure 6.2: Magnitude responses.

6.4 Some basic arithmetic

The ratios 1,10,100, 1000 give dB values 0 dB, 10 dB, 20 dB and 30 dB respectively. This implies that an increase of 10 dB corresponds to a ratio increase by a factor 10.

This can easily be shown: Given a ratio R we have R[dB] = 10 log R. Increasing the ratio by a factor of 10 we have: 10 log (10*R) = 10 log 10 + 10 log R = 10 dB + R dB.

Another important dB-value is 3dB. This comes from the fact that:

An increase by a factor 2 gives: an increase of 10 log 2 $\approx$ 3 dB. A “increase” by a factor 1/2 gives: an “increase” of 10 log 1/2 $\approx$ -3 dB.

Example 6.3
In filter terminology the **cut-off frequency** is a term that often appears. The cutoff frequency (for lowpass and highpass filters (Chapter 7)), f_c, is the frequency at which the squared magnitude response in dB is $\frac{1}{2}$. In decibel scale this corresponds to about -3 dB.

6.5 Decibels in linear systems

In signal processing we have the following relations for linear systems:

$$Y(jf) = H(jf)\,X(jf) \tag{6.5}$$

where X and H denotes the input signal and the filter respectively. Taking absolute values on both sides of (6.5) and converting to decibels we get:

NOTE: The output amplitude at a given frequency is simply given by the sum of the filter gain and the input amplitude, both in dB.

6.6 Other references:

Above we have used $P_{\text{ref}} = 1W$ as a reference and obtained the standard dB measure. In some applications it is more useful to use $P_{\text{ref}} = 1mW$ and we then have the dBm measure.

Another example is when calculating the gain of different antennas. Then it is customary to use an isotropic (equal radiation in all directions) antenna as a reference. So for a given antenna we can use the dBi measure. (i -> isotropic)

6.7 Matlab files

filter_example.m[2]

[2] http://cnx.rice.edu/content/m12452/latest/filter_example.m

Chapter 7

Filter types[1]

So what is a filter? In general a filter is a device that discriminates, according to one or more attributes at its input, what passes through it. One example is the colour filter which absorbs light at certain wavelengths. Here we shall describe frequency-selective filters. It is called freqency-selective because it discriminates among the various **frequency compononents** of its input. By filter design we can create filters that pass signals with frequency components in some bands, and attenuates signals with content in other frequency bands.

It is customary to classify filters according to their frequency domain charachteristics. In the following we will take a look at: lowpass, highpass, bandpass, bandstop, allpass and notch filters. (All of the filters shown are discrete-time)

7.1 Ideal filter types

7.1.1 Lowpass

Attenuates frequencies above cutoff frequency, letting frequencies below cutoff(f_c) through, see Figure 7.1.

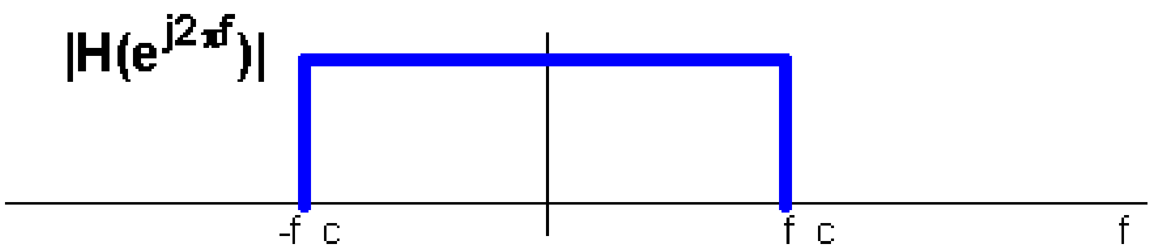


Figure 7.1: An ideal lowpass filter.

7.1.2 Highpass

Highpass filters stops low frequencies, letting higher frequencies through, see Figure 7.2.

[1] This content is available online at <http://cnx.org/content/m11868/1.6/>.

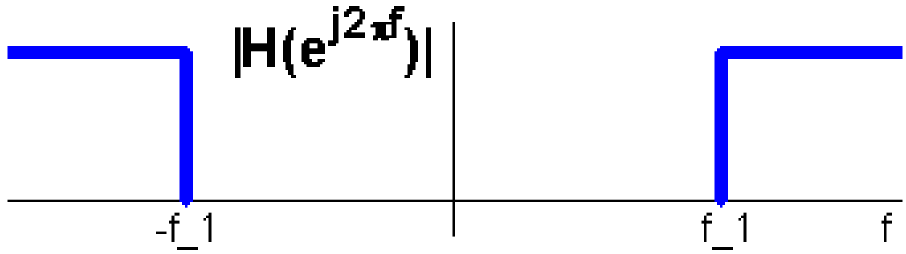


Figure 7.2: An ideal highpass filter.

7.1.3 Bandpass

Letting through only frequencies in a certain range, see Figure 7.3.

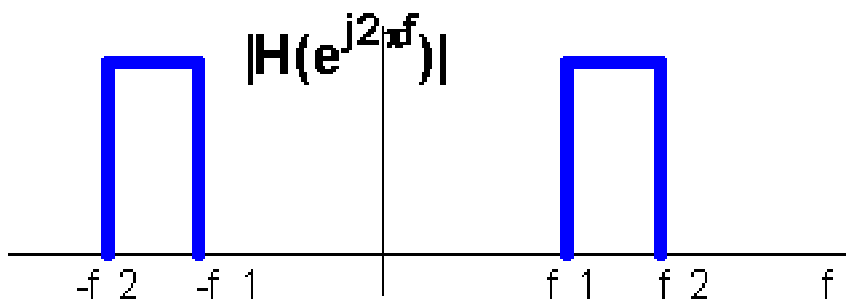


Figure 7.3: An ideal bandpass filter.

7.1.4 Bandstop

Stopping frequencies in a certain range, see Figure 7.4.

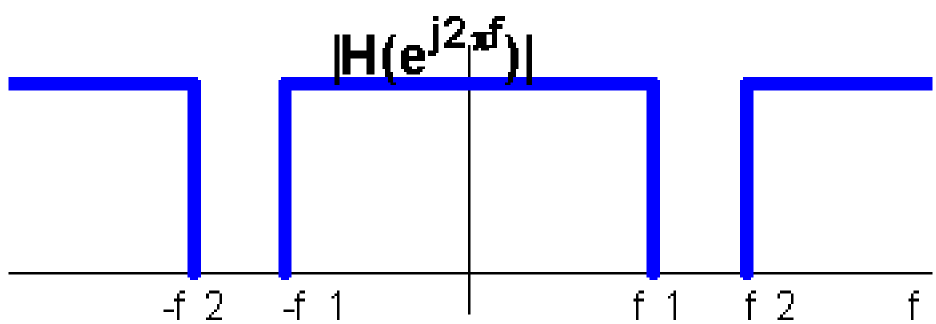


Figure 7.4: An ideal bandstop filter.

7.1.5 Allpass

Letting all frequencies through, see see Figure 7.5.

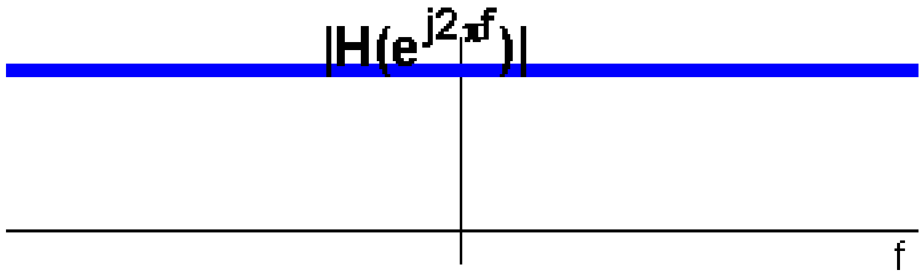


Figure 7.5: An ideal allpass filter.

Does this imply that the allpass filter is useless? The answer is no, because it may have effect on the signals phase. A filter is allpass if $|H\left(e^{j2\pi f}\right)| = 1$, $\forall f : (f)$. The allpass filter finds further applications as building blocks for many higher order filters.

7.2 Other filter types

7.2.1 Notch filter

The notch filter recognized by its perfect nulls in the frequency response, see Figure 7.6.

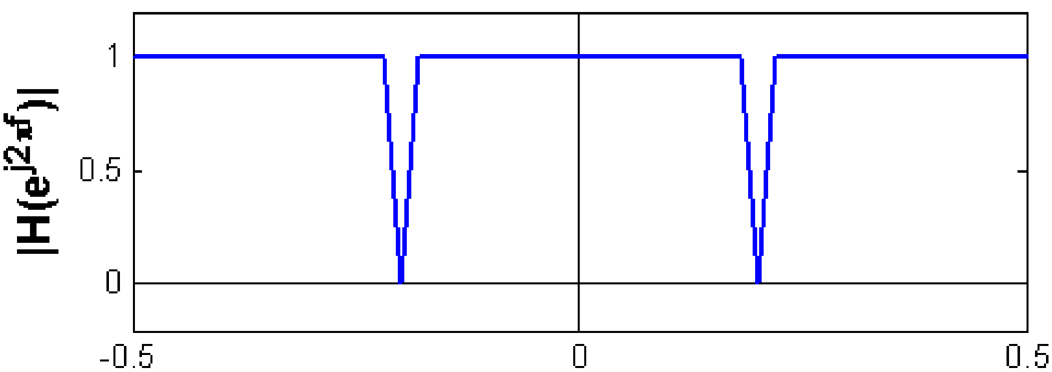


Figure 7.6: Notch filter.

Notch filters have many applications. One of them is in recording systems, where the notch filter serve to remove the power-line frequency 50 Hz and its harmonics(100 Hz, 150 Hz,...). Some audio equalisers include a notch filter.

7.3 Matlab files

idealFilters.m[2] , notchFilter.m[3]

[2]http://cnx.org/content/m11868/latest/idealFilters.m
[3]http://cnx.org/content/m11868/latest/notchFilter.m

Chapter 8

Table of Formulas[1]

Analog	**Time Discrete**
Delta function	Unit sample
$\delta(t) = 0$ for $t \neq 0$, $\int_{-\infty}^{\infty} \delta(t)\,dt = 1$	$\delta(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases}$
Unit step function	Unit step function
$u(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$	$u(n) = \begin{cases} 1 & \text{if } n \geq 0 \\ 0 & \text{otherwise} \end{cases}$
Angular frequency	Angular frequency
$\Omega = 2\pi F$	$\omega = 2\pi f$
Energy $E_a = \int_{-\infty}^{\infty} (\lvert x(t)\rvert)^2 dt$	Energy $E_d = \sum_{n=-\infty}^{\infty} (\lvert x(n)\rvert)^2$
Power $P_a = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} (\lvert x(t)\rvert)^2 dt$	Power $P_d = \frac{1}{N} \sum_{n=N_1}^{N_1+N-1} (\lvert x(n)\rvert)^2$
Convol. $y(t) = \int_{-\infty}^{\infty} x(\tau)\, h(t-\tau)\, d\tau$	Convol. $y(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k)$
Fourier Transformation	Discrete Time Fourier Transform
$X(j\Omega) = \int_{-\infty}^{\infty} x(t)\, e^{-(j\Omega t)} dt$	$X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-(j\omega n)}$
Inverse Fourier Transform	Inverse DTFT
$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega)\, e^{j\Omega t} d\Omega$	$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X\left(e^{j\omega}\right) e^{j\omega n} d\omega$
Fourier coeffecients	Discrete Fourier Transform
$\alpha_k = \frac{1}{T_0} \int_0^{T_0} x(t)\, e^{-(jk\Omega_0 t)} dt$	$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-\left(j\frac{2\pi}{N}kn\right)}$
Series expansion	Inverse DFT
$x(t) = \sum_{k=-\infty}^{\infty} \alpha_k e^{jk\Omega_0 t}$	$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, e^{j\frac{2\pi}{N}kn}$

Table 8.1

Chapter 9

Library[1]

What follows is a collection of links to other Signal processing and Information theory resources avaliable. Please report dead links and suggestions to links that we should include.

In addition to these links you should try the Connexions search function[2] which allows you to search through all the material in the Connexions system.

9.1 Signal processing

Fundamentals of Electrical Engineering[3] . A comprehensive course availiable in Roadmap/Connexions.

Signals and Systems[4] . A comprehensive course availiable in Roadmap/Connexions.

Complex to Real[5] Basic concepts, Fourier Analysis, ISI, Eye diagram...

Johns Hopkins University: Signals, Systems and Control Demonstrations. Signal Processing Tutorial[6] An **impressive** collection of Java Applets demonstrating various concepts. Recommended.

Java Digital Signal Processing Editor[7] . The J-DSP Editor, the first on-line DSP editor, is used to simulate various DSP techniques. The simulation is performed at a high level which gives the "big picture".

IEEE Signal Processing Society[8] .

9.2 Information Theory

Information Theory, Inference, and Learning Algorithms[9] . Free book by David MacKay of University of Cambridge.

A short course in Information Theory[10] , by David MacKay of University of Cambridge.

IEEE Information Theory Society [11] .

[1]This content is available online at <http://cnx.org/content/m11787/1.2/>.
[2]http://cnx.rice.edu/content/search
[3]http://cnx.rice.edu/content/col10040/1.5/
[4]http://cnx.rice.edu/content/col10064/1.4/
[5]http://www.complextoreal.com
[6]http://www.jhu.edu/∼signals/
[7]http://www.eas.asu.edu/∼midle/jdsp/
[8]http://www.ieee.org/organizations/society/sp/index.html
[9]http://www.inference.phy.cam.ac.uk/mackay/itila/book.html
[10]http://http://www.inference.phy.cam.ac.uk/mackay/info-theory/course.html
[11]http://golay.uvic.ca/

Glossary

D Differential entropy

The differential entropy $h(X)$ of a continuous random variable X with a pdf $f(x)$ is defined as

$$h(X) = -\int_{-\infty}^{\infty} f(x) \log f(x)\, dx \qquad (5.9)$$

E Entropy

1. The entropy (average self information) of a discrete random variable X is a function of its probability mass function and is defined as

$$H(X) = -\sum_{i=1}^{N} p_X(x_i) \log p_X(x_i) \qquad (5.6)$$

   where N is the number of possible values of X and $P_X(x_i) = Pr[X = x_i]$. If log is base 2 then the unit of entropy is bits per (source)symbol. Entropy is a measure of uncertainty in a random variable and a measure of information it can reveal.

2. If symbol has zero probability, which means it never occurs, it should not affect the entropy. Letting $0 \times \log 0 = 0$, we have dealt with that.

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

Attributions

Collection: *Information and Signal Theory*
Edited by: Anders Gjendemsjø
URL: http://cnx.org/content/col10211/1.19/

Module: "Cover Page"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11441/1.19/
Page: 1

Module: "Introduction"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11479/1.9/
Pages: 3-6

Module: "Discrete time signals"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11476/1.16/
Pages: 6-9

Module: "Analog signals"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11478/1.8/
Pages: 9-11

Module: "Discrete vs Analog"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11527/1.10/
Pages: 11-14

Module: "Frequency definitions and periodicity"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11477/1.13/
Pages: 15-17

Module: "Energy and Power"
By: Anders Gjendemsjø, Melissa Selik, Richard Baraniuk
URL: http://cnx.org/content/m11526/1.20/
Pages: 17-20

Module: "Exercises"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11482/1.10/
Page: 20

Module: "Introduction to Convolution"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11542/1.3/
Page: 23

Module: "Discrete Time Convolution"
By: Ricardo Radaelli-Sanchez, Richard Baraniuk, Stephen Kruzick, Catherine Elder
URL: http://cnx.org/content/m10087/2.27/
Pages: 23-29

Module: "Convolution - Analog"
By: Anders Gjendemsjø, Richard Baraniuk
URL: http://cnx.org/content/m11540/1.7/
Pages: 29-32

Module: "Convolution - Complete example"
By: Anders Gjendemsjø, Richard Baraniuk
URL: http://cnx.org/content/m11541/1.7/
Pages: 32-36

Module: "Properties of Continuous Time Convolution"
By: Melissa Selik, Richard Baraniuk, Stephen Kruzick
URL: http://cnx.org/content/m10088/2.17/
Pages: 36-39

Module: "Frequency response from a circuit diagram"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m13646/1.2/
Pages: 41-43

Module: "Introduction"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11419/1.29/
Pages: 45-47

Module: "Proof"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11423/1.27/
Pages: 47-49

Module: "Illustrations"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11443/1.33/
Pages: 49-53

Module: "Sampling and reconstruction with Matlab"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11549/1.9/
Page: 53

Module: "Aliasing Applet"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11448/1.14/
Pages: 53-54

Module: "Hold operation"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11458/1.10/
Pages: 54-56

Module: "Systems view of sampling and reconstruction"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11465/1.20/
Pages: 56-58

Module: "Exercises"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11442/1.16/
Pages: 58-59

Module: "Introduction"
By: Anders Gjendemsjø, Behnaam Aazhang
URL: http://cnx.org/content/m11838/1.5/
Pages: 63-65
Copyright: Anders Gjendemsjø, Behnaam Aazhang
License: http://creativecommons.org/licenses/by/1.0

Module: "Information"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11841/1.2/
Pages: 65-66
Copyright: Anders Gjendemsjø
License: http://creativecommons.org/licenses/by/1.0

Module: "Representing symbols by bits"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11869/1.2/
Pages: 66-68
Copyright: Anders Gjendemsjø
License: http://creativecommons.org/licenses/by/1.0

Module: "Entropy"
By: Anders Gjendemsjø, Behnaam Aazhang
URL: http://cnx.org/content/m11839/1.4/
Pages: 68-69
Copyright: Anders Gjendemsjø, Behnaam Aazhang
License: http://creativecommons.org/licenses/by/1.0

Module: "Differential Entropy"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11840/1.3/
Pages: 70-71
Copyright: Anders Gjendemsjø
License: http://creativecommons.org/licenses/by/1.0

Module: "Huffman Coding"
By: Behnaam Aazhang
URL: http://cnx.org/content/m10176/2.10/
Pages: 71-72
Copyright: Behnaam Aazhang
License: http://creativecommons.org/licenses/by/1.0

Module: "Decibel scale with signal processing applications"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m12452/1.9/
Pages: 73-76
Copyright: Anders Gjendemsjø
License: http://creativecommons.org/licenses/by/1.0

Module: "Filter types"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11868/1.6/
Pages: 77-80
Copyright: Anders Gjendemsjø
License: http://creativecommons.org/licenses/by/1.0

Module: "Table of Formulas"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11450/1.22/
Pages: 82-83

Module: "Library"
By: Anders Gjendemsjø
URL: http://cnx.org/content/m11787/1.2/
Page: 85

Information and Signal Theory
This course gives the student a broad introduction to basics in signal processing and information theory.

About Connexions
Since 1999, Connexions has been pioneering a global system where anyone can create course materials and make them fully accessible and easily reusable free of charge. We are a Web-based authoring, teaching and learning environment open to anyone interested in education, including students, teachers, professors and lifelong learners. We connect ideas and facilitate educational communities.

Connexions's modular, interactive courses are in use worldwide by universities, community colleges, K-12 schools, distance learners, and lifelong learners. Connexions materials are in many languages, including English, Spanish, Chinese, Japanese, Italian, Vietnamese, French, Portuguese, and Thai. Connexions is part of an exciting new information distribution system that allows for **Print on Demand Books**. Connexions has partnered with innovative on-demand publisher QOOP to accelerate the delivery of printed course materials and textbooks into classrooms worldwide at lower prices than traditional academic publishers.