Multidimensional digital signal processing

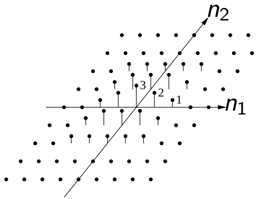
Two-dimensional discrete signals

A 2-D discrete signal (also called a **sequence** or **array**) is a function defined over the set of ordered pairs of integers:

$$x = \{x(n_1, n_2), -\infty < n_1, n_2 < \infty\}.$$

Thus $x(n_1, n_2)$ represents the **sample** of the signal x at the point (n_1, n_2). $x(n_1, n_2)$ for noninteger n_1 or n_2 is undefined.

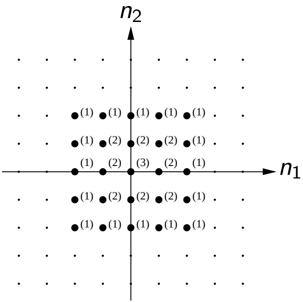
An example of a 2-D sequence is shown below:



In this figure the height at (n_1, n_2) is the amplitude of x at that point.

Representing 2-D sequences

A more convenient way of depicting a 2-D sequence is



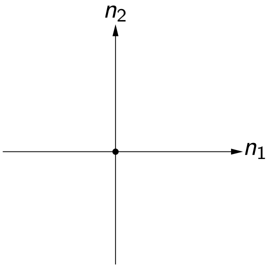
The sequence values are assumed to be zero at the values not marked with circles. A circle with no amplitude indicated represents a unit sample.

Impulse

Certain sequences and classes of sequences play an important role in 2-D signal processing. The **impulse** or unit sample sequence, denoted by $\delta(n_1, n_2)$, is defined as

$$\delta(n_1, n_2) = \begin{cases} 1, & n_1 = n_2 = 0 \\ 0, & \text{otherwise,} \end{cases}$$

and is depicted as



Decomposition in terms of impulses

Any sequence $x(n_1, n_2)$ can be represented as a linear combination of shifted impulses as follows:

$$\begin{aligned} x(n_1, n_2) = \cdots &+ x(-1,-1)\delta(n_1+1, n_2+1) + \\ &+ x(0,-1)\delta(n_1, n_2+1) \\ &+ x(1,-1)\delta(n_1-1, n_2+1) + \cdots \\ \cdots &+ x(-1,0)\delta(n_1+1, n_2) \\ &+ x(0,0)\delta(n_1, n_2) \\ &+ x(1,0)\delta(n_1-1, n_2) + \cdots \\ \cdots &+ x(-1,1)\delta(n_1+1, n_2-1) \\ &+ x(0,1)\delta(n_1, n_2-1) \\ &+ x(1,1)\delta(n_1-1, n_2-1) + \cdots \end{aligned}$$

Therefore

$$x(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2)\delta(n_1 - k_1, n_2 - k_2).$$

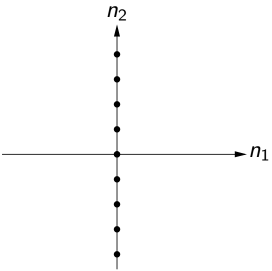
This is a convolution product, and is useful in system analysis due to the sifting property of the impulse function.

Line impulses

Line impulses have no counterpart in 1-D. An example of a line impulse is

$$x(n_1, n_2) = \delta(n_1) = \begin{cases} 1, & n_1 = 0 \\ 0, & \text{otherwise.} \end{cases}$$

This is demonstrated below:



Other examples are $\delta(n_2)$ and $\delta(n_1 - n_2)$.

Step sequence

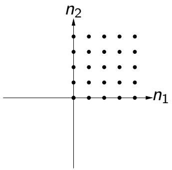
The **step sequence**,

$$u(n_1, n_2) = \begin{cases} 1, & n_1, n_2 \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

is related to $\delta(n_1, n_2)$ as follows:

$$u(n_1, n_2) = \sum_{k_1=-\infty}^{n_1} \sum_{k_2=-\infty}^{n_2} \delta(k_1, k_2).$$

This sequence is shown below:



Other examples of step sequences are $u(n_1)$, $u(n_2)$, and $u(n_1 - n_2)$, which have no counterpart in 1-D.

Exponential sequences

Exponential sequences are defined by

$$x(n_1, n_2) = a^{n_1} b^{n_2}, \qquad -\infty < n_1, n_2 < \infty,$$

where a and b are complex numbers. When a and b have unity magnitude they may be written as

$$a = e^{j\omega_1} \quad \text{and} \quad b = e^{j\omega_2},$$

in which case the exponential sequence becomes the complex sinusoidal sequence

$$\begin{aligned} x(n_1, n_2) &= e^{j\omega_1 n_1 + j\omega_2 n_2} \\ &= \cos(\omega_1 n_1 + \omega_2 n_2) + j \sin(\omega_1 n_1 + \omega_2 n_2). \end{aligned}$$

Exponential sequences are important because they are eigenfunctions of 2-D linear shift-invariant systems.

Separability

All the sequences presented thus far can be written in the form

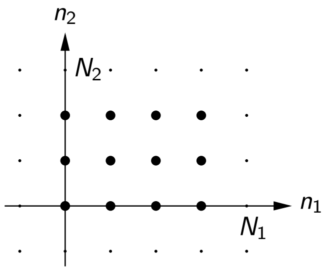
$$x(n_1, n_2) = x_1(n_1)x_2(n_2).$$

Any sequence that can be expressed as the product of 1-D sequences in this form is **separable**. Although very few actual data sequences are of this form, they are important for two reasons:

- Results for 1-D sequences that do not extend to 2-D often *do* extend to 2-D separable sequences
- Separability can often be used to reduce computation in digital filtering and transform operations.

Finite support

Finite-extent sequences are only nonzero within a finite region of support. For example, the signal



is nonzero only within the rectangle

$$0 \leq n_1 \leq N_1, \qquad 0 \leq n_2 \leq N_2.$$

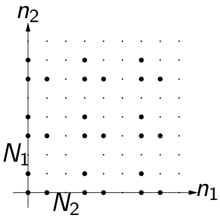
Periodicity

A **periodic sequence** in 2-D can be thought of as a sequence that repeats at regularly-spaced intervals. However, 2-D signals must repeat in two dimensions at once, so the definition is more complex than for 1-D.

A doubly periodic sequence $\tilde{x}(n_1, n_2)$ satisfies the conditions

$$\tilde{x}(n_1, n_2 + N_2) = \tilde{x}(n_1, n_2)$$
$$\tilde{x}(n_1 + N_1, n_2) = \tilde{x}(n_1, n_2).$$

Such a sequence for $N_1 = N_2 = 3$ is



General periodicity

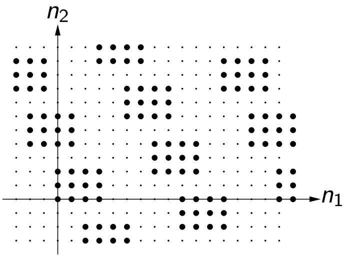
More generally, though, a periodic sequence in 2-D satisfies the conditions

$$\tilde{x}(n_1 + N_{11}, n_2 + N_{21}) = \tilde{x}(n_1, n_2)$$
$$\tilde{x}(n_1 + N_{12}, n_2 + N_{22}) = \tilde{x}(n_1, n_2),$$

where

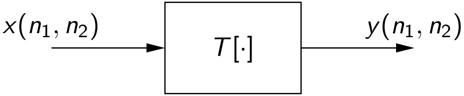
$$N_{11}N_{22} - N_{12}N_{21} = 0.$$

An example of a sequence with $N_{11} = 7$, $N_{21} = 3$, $N_{12} = -2$, $N_{22} = 4$ is



Multidimensional systems

A system is an operator that maps one signal (the input) to another (the output):



Systems can be described in terms of fundamental operations on multidimensional signals.

Fundamental operations on multidimensional sequences

Addition of two sequences $x(n_1, n_2)$ and $w(n_1, n_2)$ is defined sample-by-sample as

$$y(n_1, n_2) = x(n_1, n_2) + w(n_1, n_2).$$

Multiplication by a constant c involves multiplication by each sample value:

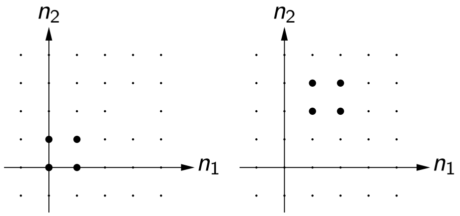
$$y(n_1, n_2) = cx(n_1, n_2).$$

Spatial shifts

A 2-D sequence x can be **linearly shifted** to form a new sequence y according to the relation

$$y(n_1, n_2) = x(n_1 - m_1, n_2 - m_2),$$

where (m_1, m_2) is the amount of the shift:



$x(n_1, n_2)$ $\quad$ $y(n_1, n_2) = x(n_1 - 1, n_2 - 2)$

Nonlinear transformations

A **spatially-varying gain** can be viewed as a generalisation of multiplication by a constant:

$$y(n_1, n_2) = c(n_1, n_2)x(n_1, n_2).$$

The collection of numbers $c(n_1, n_2)$ may also be regarded as a sequence, in which case the equation above can be interpreted as the sample-by-sample multiplication of two sequences.

Two-dimensional sequences may also be subjected to nonlinear operators. A **Memoryless nonlinearity** operator acts on each sample value of the sequence independently. For example, $y(n_1, n_2) = [x(n_1, n_2)]^2$ squares each value in the sequence x.

Linear systems

A system is **linear** if and only if the following two conditions hold:

- If the input signal is the sum of two sequences, then the output signal is the sum of the two corresponding output sequences
- Scaling the input signal produces a scaled output signal.

Linear systems obey the principle of superposition.

Linear systems

Since a general sequence x can be written as

$$x(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2)\delta(n_1 - k_1, n_2 - k_2),$$

it follows that the response of a linear system to this signal is

$$\begin{aligned} y(n_1, n_2) &= T\left[\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2)\delta(n_1 - k_1, n_2 - k_2)\right] \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2)T\left[\delta(n_1 - k_1, n_2 - k_2)\right] \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2)h_{k_1 k_2}(n_1 - k_1, n_2 - k_2), \end{aligned}$$

where $h_{k_1 k_2}(n_1 - k_1, n_2 - k_2)$ is the response of the system to a unit impulse at (k_1, k_2).

Shift-invariant systems

A **shift-invariant** system is one for which a shift in the input sequence implies a corresponding shift in the output sequence. If $y(n_1, n_2) = T[x(n_1, n_2)]$, the system T is shift invariant if and only if

$$T\left[x(n_1 - m_1, n_2 - m_2)\right] = y(n_1 - m_1, n_2 - m_2)$$

for all sequences x and linear shifts (m_1, m_2).

Linear shift-invariant systems

The spatially-varying impulse response for a linear system

$$h_{k_1 k_2}(n_1 - k_1, n_2 - k_2) = T\left[\delta(n_1 - k_1, n_2 - k_2)\right],$$

when evaluated at $k_1 = k_2 = 0$ implies that

$$h_{00}(n_1, n_2) = T[\delta(n_1, n_2)].$$

Applying the principle of shift invariance we get

$$h_{k_1 k_2}(n_1, n_2) = h_{00}(n_1 - k_1, n_2 - k_2).$$

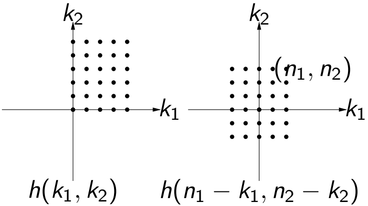
The spatially-varying impulse response becomes a shifted replica of a spatially-invariant impulse response. Defining $h(n_1, n_2) = h_{00}(n_1, n_2)$ we can write the input-output relation as

$$y(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2).$$

This is the 2-D convolution product.

Convolution in 2-D

Two-dimensional convolution is very similar to its 1-D counterpart, and there is also a computational interpretation of the convolution sum. To obtain this, we consider $x(n_1, n_2)$ and $h(n_1 - k_1, n_2 - k_2)$ as functions of k_1 and k_2. To generate the sequence $h(n_1 - k_1, n_2 - k_2)$ from $h(n_1, n_2)$, h is first reflected about both the k_1 and k_2 axes, and then translated so that the sample $h(0, 0)$ lies at the point (n_1, n_2):



The product sequence $x(k_1, k_2)h(n_1 - k_1, n_2 - k_2)$ can then be formed, and the output sample value $y(n_1, n_2)$ is computed by summing the nonzero sample values in the product sequence.

Properties of linear systems

The properties of LSI 2-D systems are analogous to those for 1-D LTI systems. For example, when 2-D LSI systems are connected in parallel or cascade, then their overall response is the sum and the convolution of the system impulse responses respectively. Also, a 2-D LSI system is stable in the BIBO sense if and only if its impulse response is absolutely summable:

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |h(n_1, n_2)| = S_1 < \infty.$$

However, one must be careful not to take the analogy too far — 2-D LSI systems are considerably more complex than 1-D LTI systems.

Separable impulse response

One property of 2-D systems that has no counterpart in 1-D is separability. A **separable** system is an LSI system whose impulse response is a separable sequence, so $h(n_1, n_2) = h_1(n_1)h_2(n_2)$. The input signals processed by a separable system and the signals produced by it need not be separable. For separable systems the convolution sum decomposes as

$$
\begin{aligned}
y(n_1, n_2) &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(n_1 - k_1, n_2 - k_2)h_1(k_1)h_2(k_2) \\
&= \sum_{k_1=-\infty}^{\infty} h_1(k_1) \sum_{k_2=-\infty}^{\infty} x(n_1 - k_1, n_2 - k_2)h_2(k_2).
\end{aligned}
$$

Separable impulse response (2)

Defining

$$g(n_1, n_2) = \sum_{k_2=-\infty}^{\infty} x(n_1, n_2 - k_2)h_2(k_2),$$

this becomes

$$y(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} h_1(k_1)g(n_1 - k_1, n_2).$$

The array $g(n_1, n_2)$ can be computed by performing 1-D convolution between each column of x (n_1 constant) and the 1-D sequence h_2. The output array y can then be computed by convolving each row of g (n_2 constant) with the 1-D sequence h_1. Thus the output can be obtained as a series of 1-D convolutions. Note that the row convolutions can also be performed before the column convolutions.

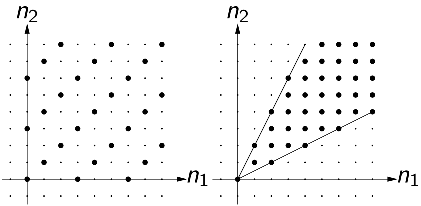
Region of support

Another difference between 1-D and 2-D systems involves **regions of support** of signals. In 1-D it was useful to characterise a system as causal if the outputs did not preceed the inputs. For most 2-D applications the independent variables do not correspond to time, and causality is not a natural constraint.

The impulse response $h[n]$ of a 1-D causal LTI system is zero for $n < 0$. One generalisation of the concept of causality for 2-D systems can be made by requiring that the impulse response be zero outside of some region of support.

Region of support (2)

Two common regions of support are:



Quadrant support Wedge support

- **Quadrant support**, where sequences are nonzero only in one quadrant of the (n_1, n_2)-plane
- **Wedge support**, which is a generalisation of quadrant support, and implies that the sequence is only nonzero inside a sector defined by two rays emanating from the origin.

Different regions of support lead to systems with different characteristics.

Signals and systems in the frequency domain

Complex sinusoidal sequences of the form

$$x(n_1, n_2) = e^{j\omega_1 n_1 + j\omega_2 n_2}$$

are eigenfunctions of 2-D LSI systems:

$$\begin{aligned} y(n_1, n_2) &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} e^{j\omega_1(n_1-k_1)+j\omega_2(n_2-k_2)} h(k_1, k_2) \\ &= e^{j\omega_1 n_1 + j\omega_2 n_2} H(\omega_1, \omega_2), \end{aligned}$$

where $H(\omega_1, \omega_2)$ is the **frequency response**

$$H(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} h(n_1, n_2) e^{-j\omega_1 n_1 - j\omega_2 n_2}.$$

This function is periodic in both the horizontal and vertical frequency variables:

$$\begin{aligned} H(\omega_1 + 2\pi, \omega_2) &= H(\omega_1, \omega_2), \\ H(\omega_1, \omega_2 + 2\pi) &= H(\omega_1, \omega_2). \end{aligned}$$

Multidimensional Fourier transform

The **multidimensional Fourier transform** analysis equation is therefore

$$X(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2) e^{-j\omega_1 n_1 - j\omega_2 n_2}.$$

The corresponding synthesis equation is

$$x(n_1, n_2) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} X(\omega_1, \omega_2) e^{j\omega_1 n_1 + j\omega_2 n_2} d\omega_1 d\omega_2.$$

The Fourier transform can be shown to exist whenever the sequence $x(n_1, n_2)$ is absolutely summable.

Fourier transform properties

The properties of the Fourier transform are similar to those in 1-D, with the exception of some added complexity due to the introduction of an additional parameter. For example, the time-reversal property in 1-D becomes a more general reflection property in 2-D:

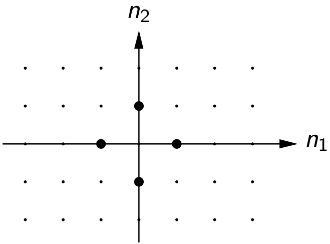
$$x(-n_1, n_2) \xleftrightarrow{\mathcal{F}} X(-\omega_1, \omega_2)$$

$$x(n_1, -n_2) \xleftrightarrow{\mathcal{F}} X(\omega_1, -\omega_2)$$

$$x(-n_1, -n_2) \xleftrightarrow{\mathcal{F}} X(-\omega_1, -\omega_2).$$

Example: Frequency response of a simple system

The frequency response of the system with impulse response



is given by

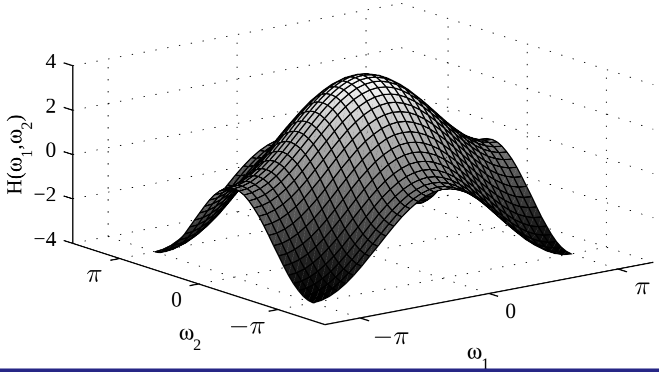
$$H(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} [\delta(n_1+1, n_2) + \delta(n_1-1, n_2) + \delta(n_1, n_2+1) + \delta(n_1, n_2-1)] e^{-j\omega_1 n_1 - j\omega_2 n_2}.$$

Example: Frequency response of a simple system (2)

This evaluates to

$$H(\omega_1, \omega_2) = e^{j\omega_1} + e^{-j\omega_1} + e^{j\omega_2} + e^{-j\omega_2} = 2(\cos\omega_1 + \cos\omega_2).$$

and is shown below:

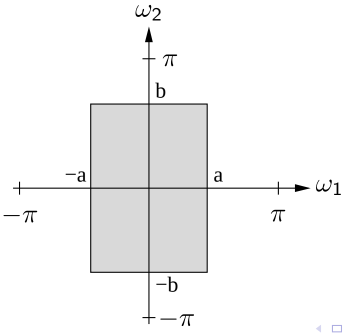


Example: Separable ideal lowpass filter

Consider the ideal lowpass filter specified by the frequency response

$$H(\omega_1, \omega_2) = \begin{cases} 1, & |\omega_1| \leq a < \pi, |\omega_2| \leq b < \pi \\ 0, & \text{otherwise,} \end{cases}$$

indicated in the figure below:



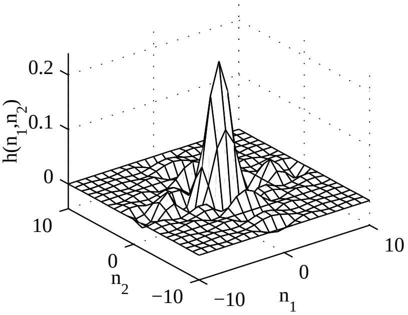
Example: Separable ideal lowpass filter (2)

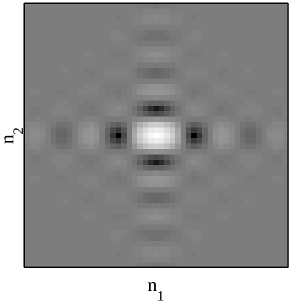
Since this system is separable the inverse Fourier transform is quite simple:

$$
\begin{aligned}
h(n_1, n_2) &= \frac{1}{4\pi^2} \int_{-a}^{a} \int_{-b}^{b} e^{j\omega_1 n_1 + j\omega_2 n_2} d\omega_2 d\omega_1 \\
&= \left(\frac{1}{2\pi} \int_{-a}^{a} e^{-j\omega_1 n_1} d\omega_1 \right) \left(\frac{1}{2\pi} \int_{-b}^{b} e^{-j\omega_2 n_2} d\omega_2 \right) \\
&= \frac{\sin(an_1)}{\pi n_1} \frac{\sin(bn_2)}{\pi n_2}.
\end{aligned}
$$

Example: Separable ideal lowpass filter (3)

This function is depicted in the surface plot and image below, for the case of $a = 0.4\pi$ and $b = 0.6\pi$. The gray level at any point in the image is proportional to the function value:



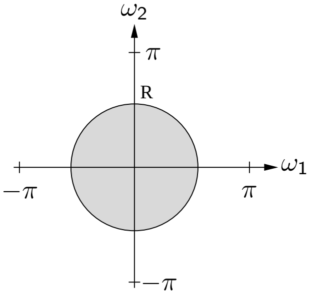


Example: Nonseparable lowpass filter

As a more complex example, consider the problem of determining the impulse response of the ideal circular lowpass filter

$$H(\omega_1, \omega_2) = \begin{cases} 1, & \omega_1^2 + \omega_2^2 \leq R^2 < \pi^2 \\ 0, & \text{otherwise.} \end{cases}$$

This frequency response, which is not separable, is shown below:



The impulse response is given by

$$h(n_1, n_2) = \frac{1}{4\pi^2} \iint_A e^{j\omega_1 n_1 + j\omega_2 n_2} d\omega_1 d\omega_2,$$

where A is the shaded area in the above figure. Defining

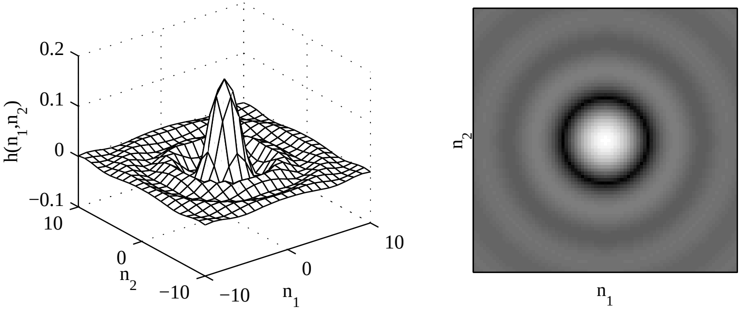
$$\omega = \sqrt{\omega_1^2 + \omega_2^2}, \qquad \phi = \tan^{-1}\frac{\omega_2}{\omega_1}, \qquad \theta = \tan^{-1}\frac{n_1}{n_1},$$

the impulse response becomes

$$\begin{aligned} h(n_1, n_2) &= \frac{1}{4\pi^2} \int_0^R \int_0^{2\pi} \omega e^{j\omega\sqrt{n_1^2+n_2^2}\cos(\theta-\phi)} d\phi d\omega \\ &= \frac{1}{2\pi} \int_0^R \omega J_0\left(\omega\sqrt{n_1^2 + n_2^2}\right) d\omega \\ &= \frac{R}{2\pi} \frac{J_1\left(R\sqrt{n_1^2 + n_2^2}\right)}{\sqrt{n_1^2 + n_2^2}} \end{aligned}$$

where J_0 and J_1 are Bessel functions of the first kind of orders 0 and 1 respectively.

This function is shown below for $R = 0.5\pi$:



In general, the 2-D Fourier transform of a rotationally-symmetric function is itself rotationally symmetric: the 1-D profiles are related according to the Hankel transform

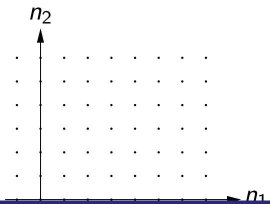
$$F(q) = 2\pi \int_0^\infty f(r) J_0(2\pi q r) r dr.$$

Sampling

The most common way of representing a continuous signal is in terms of a discrete signal obtained by periodic sampling. The simplest way of generalising 1-D periodic sampling to the 2-D case is by using **rectangular sampling**, where periodic sampling is done in rectangular coordinates. If $x_a(t_1, t_2)$ is a continuous-time waveform, then the discrete signal obtained by rectangular sampling is

$$x(n_1, n_2) = x_a(n_1 T_1, n_2 T_2),$$

where T_1 and T_2 are the horizontal and vertical sampling intervals. This corresponds to samples in the plane:

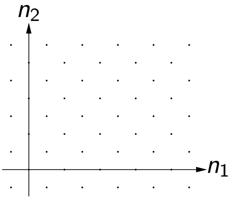


The criterion for being able to reconstruct x_a from the samples is that T_1 and T_2 be chosen small enough that

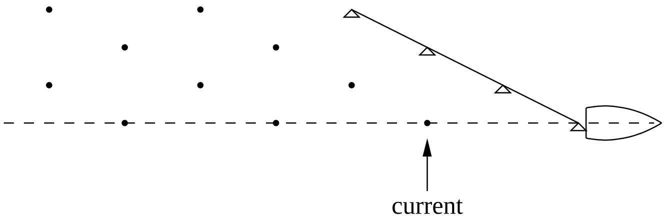
$$X_a(\Omega_1, \Omega_2) = 0 \qquad \text{for } |\Omega_1| \geq \frac{\pi}{T_1}, |\Omega_2| \geq \frac{\pi}{T_2}.$$

Thus the signal must be bandlimited in the 2-D Fourier domain. If this condition is not met, aliasing results in the same way as in 1-D.

Rectangular sampling is not the only option for periodic sampling. For circularly-symmetric bandlimited signals it can be shown that there is no more efficient sampling scheme than *hexagonal sampling*. That is, such signals can be represented by fewer sample points than with any other sampling geometry. Hexagonal sampling is performed on a grid such as



Sometimes the manner in which data are acquired determines the sampling pattern. For example, in the preparation of seismic maps it is common for a boat to tow a uniform microphone array while covering an area. The presence of a cross current can cause an offset in the angle of the array, resulting in unusual sampling:



If a signal is sampled in such a way that aliasing is avoided, it is in general possible to transform the representation onto the sampling grid of choice. In principle one just needs to reconstruct the signal, and resample it as required.

The multidimensional discrete Fourier transform

The multidimensional DFT is a simple generalisation of that for 1-D: the forward transform is given by

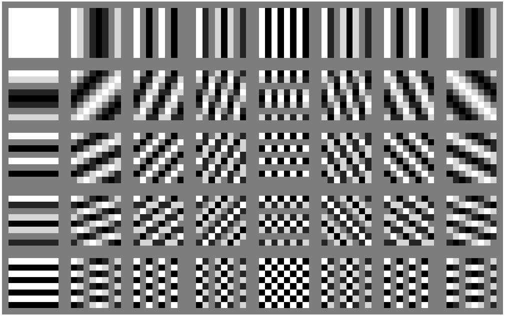
$$X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) e^{-j2\pi k_1 n_1/N_1 - j2\pi k_2 n_2/N_2}$$

and the inverse transform by

$$x(n_1, n_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} X(k_1, k_2) e^{j2\pi k_1 n_1/N_1 + j2\pi k_2 n_2/N_2}$$

The first equation formally holds for $0 \le k_1 \le N_1 - 1$, $0 \le k_2 \le N_2 - 1$ and the second for $0 \le n_1 \le N_1 - 1$, $0 \le n_2 \le N_2 - 1$.

The real part of some of the basis functions for the 2-D DFT are



If the sequence $x(n_1, n_2)$ is only supported on $0 \leq n_1 \leq N_1 - 1$, $0 \leq n_2 \leq N_2 - 1$ (and is therefore zero outside of this area), then the DFT consists of samples of the 2-D Fourier transform

$$X(k_1, k_2) = X(\omega_1, \omega_2)\big|_{\omega_1 = 2\pi k_1/N_1, \omega_2 = 2\pi k_2/N_2}.$$

The properties of the 2-D DFT are similar to those in 1-D. In particular, the signals obtained from both the DFT and the inverse DFT are implicitly doubly periodic with period N_1 in the horizontal direction and N_2 in the vertical direction. Circular convolution according to the relation

$$y(n_1, n_2) = \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} h(m_1, m_2) x(((n_1 - m_1))_{N_1}, ((n_2 - m_2))_{N_2})$$

is therefore implicit in all shifts involved in the DFT properties.

A direct implementation of the DFT requires $N_1^2 N_2^2$ complex multiplications and additions. However, fast Fourier transforms can also be developed in multiple dimensions, and in 2-D require of the order of $N_1 N_2 \log_2 N_1 N_2$ operations. These methods rely on a row-column decomposition of the DFT sum:

$$X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \left[\sum_{n_2=0}^{N_2-1} x(n_1, n_2) e^{-j2\pi k_2 n_2 / N_2} \right] e^{-j2\pi k_1 n_1 / N_1}.$$

Thus the 2-D DFT can be calculated by performing 1-D DFTs on each column of x, followed by 1-D DFTs on each row. Fast algorithms for 1-D DFTs can therefore be used to develop multidimensional FFT methods.

Other extensions of linear system theory

Aspects of linear system theory for 1-D signals carry over to multiple dimensions, but often have very different properties.

LSI systems are generally implemented using difference equations. In the 2-D case a difference equation takes the form

$$\sum_{k_1=0}^{N_1}\sum_{k_2=0}^{N_2} b(k_1,k_2)y(n_1-k_1,n_2-k_2) = \sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2} a(r_1,r_2)x(n_1-r_1,n_2-r_2).$$

However, although multidimensional difference equations represent a generalisation of 1-D difference equations, they are considerably more complex, and are in fact quite different. For example, issues such as stability are far more difficult to understand for higher-dimensional systems.

A two-dimensional z-transform can be defined according to

$$H(z_1, z_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1, k_2) z_1^{-k_1} z_2^{-k_2}.$$

This transform is very different in its 2-D form: poles and zeros form continuous surfaces in a 4-D space. Also, it is generally not possible to factorise a 2-D polynomial (there is no fundamental theorem of algebra for multidimensional polynomials). Nonetheless, the multidimensional z-transform has been exhaustively analysed in the literature, and plays an important role in the understanding of multidimensional systems.

The principles and methods of FIR filter design in 1-D extend naturally to 2-D. However, since causality is seldom an issue, a useful linear phase condition is the zero phase condition

$$H(\omega_1, \omega_2) = H^*(\omega_1, \omega_2).$$

A linear phase response for digital filters is important to many applications in multidimensional DSP. For example, in image processing a nonlinear phase response tends to destroy lines and edges:

Original image

Linear-phase lowpass filter

Nonlinear-phase lowpass filter

There are multidimensional counterparts to the window method of filter design, as well as the frequency-sampling method, the optimal method, and most others. Also, since FIR filters are specified in terms of convolutions, the multidimensional FFT can be used in the implementation. This becomes extremely important in high dimensions, due to the vast amount of data involved.

Shown below are examples of filter responses in 2-D: