9 Fourier Transform of a periodic function

9.1 FT of periodic function

DEMO

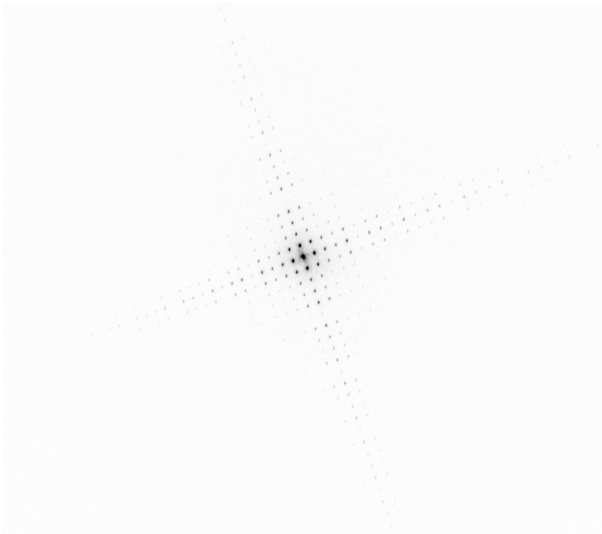


Figure 53:

Any periodic function can be described by a Fourier series

$$f_p(t) = \sum_{n=-\infty}^{\infty} F_n e^{jn\omega_0 t}$$

Where ω_0 is the fundamental frequency.
Now we would like to find :

$$\begin{aligned}
F(\omega) &= \mathcal{F}\left\{f_p(t)\right\} = \int\limits_{-\infty}^{\infty} \left(\sum_{n=-\infty}^{\infty} F_n e^{jn\omega_0 t} \right) e^{-j\omega t} \mathrm{d}t = \\
&= \sum_{n=-\infty}^{\infty} F_n \int\limits_{-\infty}^{\infty} e^{jn\omega_0 t} e^{-j\omega t} \mathrm{d}t = \sum_{n=-\infty}^{\infty} F_n \mathcal{F}\left\{e^{jn\omega_0 t}\right\} \\
\mathcal{F}\left\{\delta(t-t_0)\right\} &= e^{-j\omega t} \\
\mathcal{F}\left\{e^{j\omega_0 t}\right\} &= 2\pi\delta(\omega-\omega_0) \\
F(\omega) &= \sum_{n=-\infty}^{\infty} 2\pi F_n \delta(\omega - n\omega_0)
\end{aligned}$$

F_n are found as usual; here $f(t)$ is not zero only between $-\frac{T_0}{2}$ and $+\frac{T_0}{2}$. Whole periodic function is defined by

$$f_p(t) = f(t) * \delta_T(t)$$
$$\delta_T = \sum_{n=-\infty}^{\infty} \delta(t - nT_0)$$

$$F_n = \frac{1}{T_0}\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} f(t)e^{-jn\omega_0 t} = \frac{1}{T_0}\int_{-\infty}^{\infty} f(t)e^{-j\omega t}\Bigg|_{w=n\omega_0} = \frac{1}{T_0}F(n\omega_0)$$

Since $f(t)$ is zero outsize the 1st period, we can change the integration limits. Further more, we are not interested in general transform of the $f(t)$ function, but in F_n which are equivalent to transform calculated at frequency $\omega_0 n$:

$$F(w) = \sum_{n=-\infty}^{\infty}\frac{2\pi}{T_0}F(n\omega_0)\delta(w-n\omega_0) = \sum_{n=-\infty}^{\infty}\omega_0 F(n\omega_0)\delta(w-n\omega_0)$$

◇ **Example 34.** FT of pulse train. Rectangular pulse with an amplitude A, duration τ and repetition period T_0.

$$F(\omega) = \sum_{n=-\infty}^{\infty}\frac{2\pi}{T_0}F(n\omega_0)\delta(\omega-n\omega_0)$$

Fourier transform of the pulse function $P(t)$:

$$\mathcal{F}\left\{P(t)\right\} = \mathcal{F}\left\{A\mathrm{rect}\left(\frac{\mathrm{t}}{\tau}\right)\right\} = A\tau\frac{\sin\left(\frac{\omega\tau}{2}\right)}{\frac{\omega\tau}{2}}$$

$$F(\omega) = \sum_{n=-\infty}^{\infty} A\omega_0\tau\frac{\sin\left(\frac{\omega\tau}{2}\right)}{\frac{\omega\tau}{2}}\delta(\omega-n\omega_0)$$

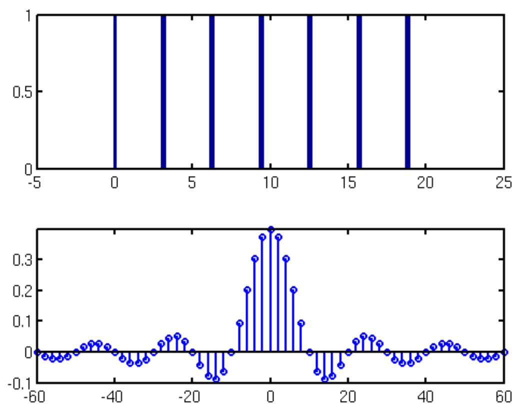


Figure 54:

◇ **Example 35.** FT of a train of impulse functions

$$
\begin{aligned}
f_p &= \sum_{n=-\infty}^{\infty} \delta(t - nT_0) \\
F(\omega) &= \sum_{n=-\infty}^{\infty} \frac{2\pi}{T_0} \mathcal{F}\{\delta(t)\} \delta(\omega - n\omega_0) = \sum_{n=-\infty}^{\infty} \frac{2\pi}{T_0} \delta\left(\omega - \frac{2\pi n}{T_0}\right) = \\
&= \sum_{n=-\infty}^{\infty} \omega_0 \delta\left(\omega - n\omega_0\right)
\end{aligned}
$$

So those are δ-functions in the frequency domain spaced by $n\frac{2\pi}{T_0}$.
The train of impulse is essential in understanding of data sampling. ♣

9.2 Data sampling

9.2.1 Ideal impulse sampling

Two problems may occur when we convert continuous time and amplitude signal to discrete time discrete amplitude (digital) signal. We might get errors related to quantization or sampling frequency.

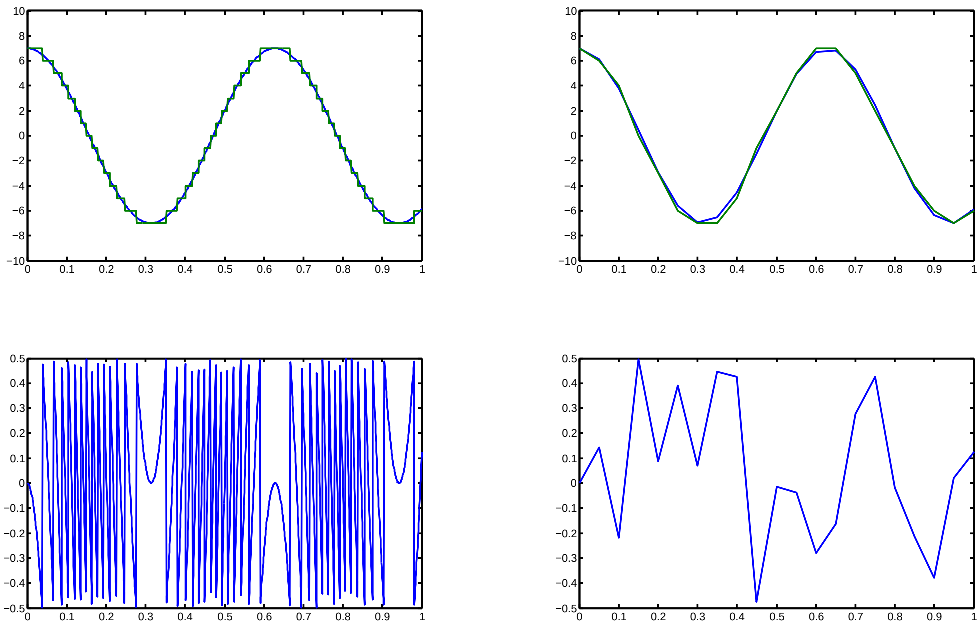


Figure 55: sampling

We can use train of sample pulses to sample our data:

$$
\delta_T = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)
$$

Where T_S is the sample period. Now our sampled function can be written as:

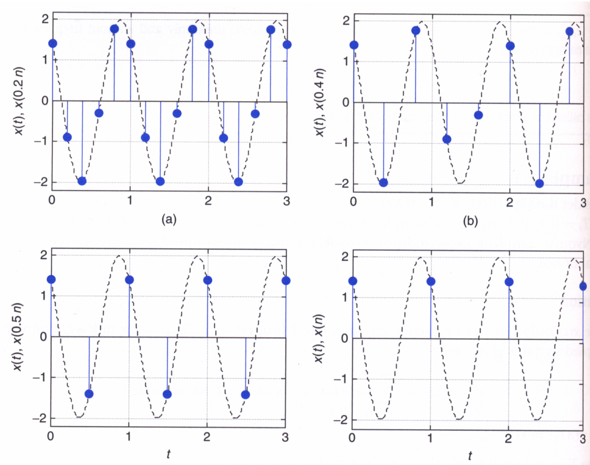


Figure 56:

$$f_s(t) = f(t)\delta_T(t)$$

Multiplication in the time domain corresponds to convolution in the frequency domain.

$$F_s(w) = \frac{1}{2\pi}F(\omega) * \mathcal{F}\left\{\sum_{n=-\infty}^{\infty}\delta(t-nT_S)\right\} =$$

$$= \frac{1}{2\pi}F(\omega) * \omega_s\sum_{n=-\infty}^{\infty}\delta(\omega-n\omega_s) = \frac{\omega_s}{2\pi}\sum_{n=-\infty}^{\infty}F(\omega-n\omega_s) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty}F(\omega-n\omega_s)$$

Definition 24 Fourier transform of time discrete function $f_s(t)$ is given by

$$F_s(w) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty}F(\omega-n\omega_s) \tag{59}$$

So the Fourier transform is just multiple copies of the $F(\omega)$ in the frequency space offset by the sampling frequency!

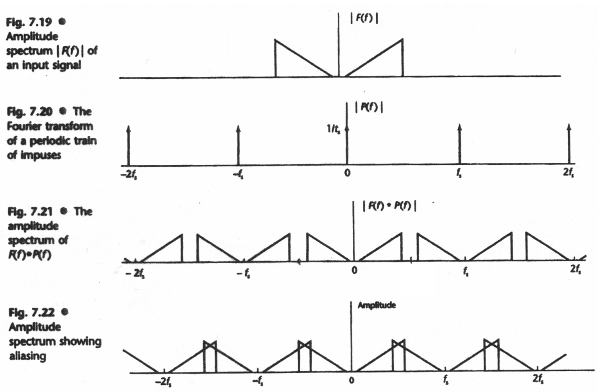


Figure 57: Aliasing

9.2.2 Aliasing

If samplig frequency is to low, this can results in "overlap" between $F(\omega)$ and its copies offset by the sampling frequency. In that case new low frequency signal will be added to original signal creating effect called "Aliasing".

Definition 25 Nyquist sampling rate condition A band limited signal $x(t)$ for which

$$|X(\omega)| = 0 \quad \text{for} \quad |\omega| > \omega_{\max} \tag{60}$$

can be sampled without frequency aliasing using a sampling frequency

$$\omega_s = \frac{2\pi}{T_s} \geq 2\omega_{max} \tag{61}$$

Some examples: `http://www0.cs.ucl.ac.uk/teaching/GZ05/samples/`

◇ **Example 36.** Lets consider a simple cosine function with frequency f_0 sampled at f_s

$$x(t) = \cos(\omega_0 t) \quad \omega_0 = 2\pi f_0 \quad f_0 = 1\text{kHZ}$$

sampling

$$\begin{aligned} p(t) &= \sum_{n=-\infty}^{\infty} \delta(t - nt_s) \quad T_s = \frac{1}{f_s} \quad f_s = 8kHz \\ x_s(t) &= p(t)x(t) \end{aligned}$$

Fourier series expansion of the sampling function:

$$\begin{aligned} F_k &= \frac{1}{T_s}\int_0^{T_s} e^{-jk\omega_s t}\delta(t)\mathrm{d}t = \frac{1}{T_s} \\ p(t) &= \sum_{n=-\infty}^{\infty} \frac{1}{T_s} e^{jn\omega_s t} = \frac{1}{T_s}\left[1 + \sum_{n=1}^{\infty}\left(e^{jn\omega_s t} + e^{-jn\omega_s t}\right)\right] \end{aligned}$$

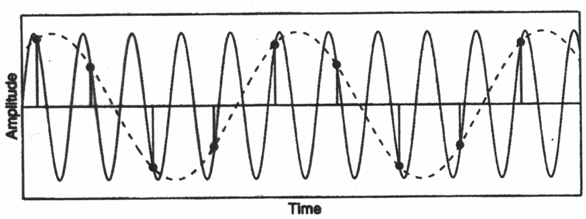


Figure 58:

Then:

$$\begin{aligned} p(t) &= \frac{1}{T_s}\sum_{n=1}^{\infty}(1+2\cos(n\omega_s t)) \\ x_s(t) &= \frac{1}{T_s}\cos(\omega_0 t)\sum_{n=1}^{\infty}(1+2\cos(n\omega_s t)) = \qquad (62) \\ &= \frac{1}{T_s}\cos(\omega_0 t)+\frac{1}{T_s}\sum_{n=1}^{\infty}[\cos(n\omega_s t+\omega_0 t)+\cos(n\omega_s t-\omega_0 t)] \qquad (63) \end{aligned}$$

♣

using:

$$\cos\alpha\cos\beta = \frac{1}{2}\cos(\alpha+\beta)+\frac{1}{2}\cos(\alpha-\beta)$$

```
t=0:1/1e3:2;
w0 = 2*pi*1;
ws = 2*pi*10;
x = cos(w0*t);
N =3;
k = 1:N;
n = zeros(N,1)+1;
k = k';
y = (cos(w0*t) +
sum(cos(ws*k*t+w0*n*t),1));
subplot(3,1,1)
plot(t,y,t,x)


N =20;
k = 1:N;
n = zeros(N,1)+1;
k = k';
y = 1/N*(cos(w0*t) +
sum(cos(ws*k*t+w0*n*t),1));
subplot(3,1,2)
plot(t,y,t,x)


N =700;
k = 1:N;
n = zeros(N,1)+1;
k = k';
y = (cos(w0*t) +
sum(cos(ws*k*t+w0*n*t),1));
subplot(3,1,1)
plot(t,y,t,x)
```

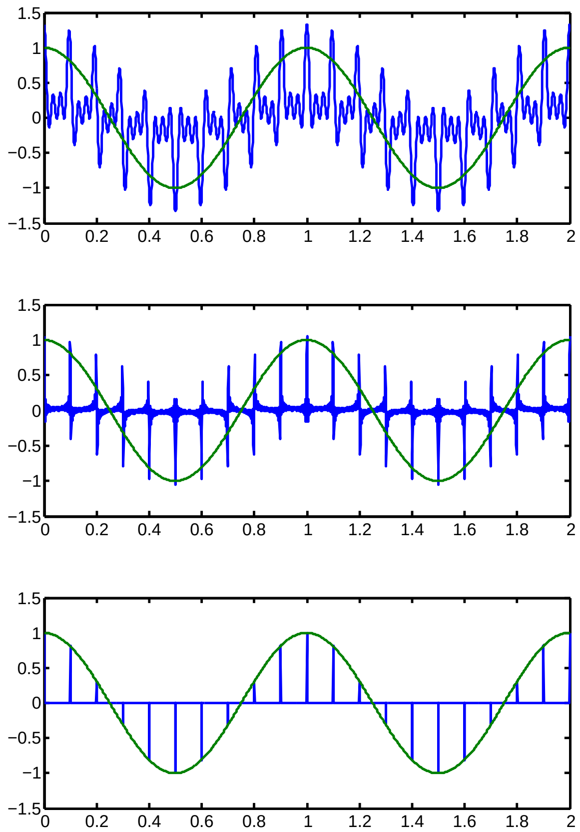


Figure 59: Sampled cos function according to eq. 63 with 3, 15 and 700 terms.

9.2.3 Reconstruction of sampled data

If we use a low-pass filter we can recover the sampled data

$$H_{lp}(jw) = \begin{cases} T_s & -\omega_s/2 < \omega < \omega_s/2 \\ 0 & \text{elswhere} \end{cases}$$

So a band-limited signal which is sampled using a sampling period T_S which satisfy Nyquist sampling rate criteria can be recovered by means of an ideal low-pass filter. Now:

$$X_r(\omega) = H_{lp}(w)X_s(\omega)$$

If we consider a signal

$$x(t) = \epsilon(t+0.5) - \epsilon(t-0.5)$$

The Fourier transform of that signal will be

$$X(\omega) = 0.5\frac{\sin(0.5\omega)}{0.5\omega}$$

Which in principle is not band limited. But we can use Parseval's energy relation to find a cut off frequency ω_{max}, such as, for example 99% of the signal energy is in the frequency band $[-\omega_{max}, \omega_{max}]$

9.2.4 Sinc interpolation

For an ideal low pass filter

$$H_{lp}(iw) = \begin{cases} T_s & -\omega_s/2 < \omega < \omega_s/2 \\ 0 & \text{elswhere} \end{cases}$$

the impulse response can be calculated from inverse Fourier transform of $H(\omega)$:

$$h_{lp}(t) = T_s\frac{1}{2\pi}\int_{-\infty}^{\infty} 1e^{j\omega t}\mathrm{d}\omega = \int_{-\omega_s/2}^{\omega_s/2} e^{j\omega t}\mathrm{d}\omega =$$

$$= \frac{T_s}{2\pi}\frac{1}{it}\left[e^{j\omega t}\right]_{\omega_s/2}^{\omega_s/2} = \frac{T_s}{2\pi}\frac{1}{i}\frac{\omega_s}{2}\left[\frac{e^{j\omega_s t/2} - e^{-j\omega_s t/2}}{t\omega_s/2}\right] =$$

$$= \operatorname{sinc}\left(\frac{2\pi \mathrm{t}}{2\mathrm{T_s}}\right) = \operatorname{sinc}\left(\frac{2\pi \mathrm{t}}{2\mathrm{T_s}}\right) = \operatorname{sinc}\left(\frac{\pi \mathrm{t}}{\mathrm{T_s}}\right)$$

Reconstruction of the signal can be obtained from a convolution between sampled signal and impulse response:

$$x_r(t) = x_s(t) * h_{lp}(t) = \int_{-\infty}^{\infty} x_s(\tau)h_{lp}(\tau - t)\mathrm{d}\tau =$$

$$= \int_{-\infty}^{\infty}\sum_{n=-\infty}^{\infty} x(nT_s)\delta(\tau - nT_s)h_{lp}(\tau - t)\mathrm{d}\tau =$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s)\operatorname{sinc}\left(\frac{\pi(\mathrm{t} - \mathrm{nT_s})}{\mathrm{T_s}}\right)$$

The recovered signal is therefore an interpolation in terms of time-shifted sinc signal with amplitudes given by the input signal at sampling time-points $x(nT_s)$. If we let $t = kT_s$ we get:

$$x_r(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\,\mathrm{sinc}\left(\frac{\pi(\mathrm{kT_s} - \mathrm{nT_s})}{\mathrm{T_s}}\right) = \sum_{\mathrm{n}=-\infty}^{\infty} \mathrm{x(nT_s)\,sinc(\pi(k-n))}$$

which is 1 for $n = k$ and zero elswhere.

9.2.5 Non-ideal Sampling

In real life, to sample the signal we have to "take out" some of its energy, for example by loading a capacitor at sample time-points. The resulting capacitor voltage is measured to give $x_s(nT_s)$. This however requires integration over time span $\tau > 0$.

$$x_s(t) = \left[x(t) * \frac{1}{\tau}\mathrm{rect}\left(\frac{\mathrm{t}}{\tau}\right)\right] \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

As:

$$\frac{1}{\tau}\int_{t-\tau/2}^{t+\tau/2} x(t')\mathrm{d}t' = x(t) * \frac{1}{\tau}\mathrm{rect}\left(\frac{\mathrm{t}}{\tau}\right)$$

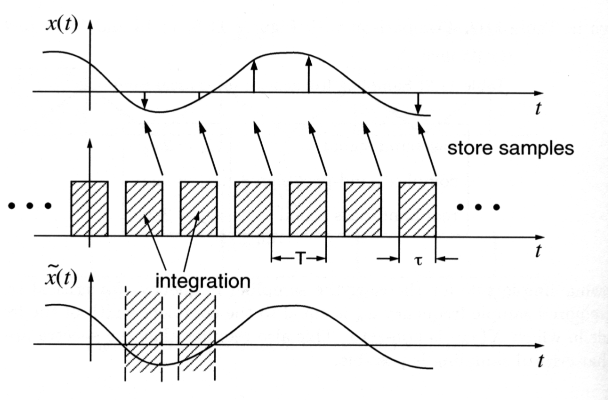


Figure 60:

10 DSP - Discrete time signal (CH12)

Such signal arise from sampling of contionous signals x(t) at intervals Y. Typically we denote such signals as

$$x[k] = x(kT_s) \quad k \text{ is integer}[-\infty, \infty]$$

where T_s is the time between successive samples.

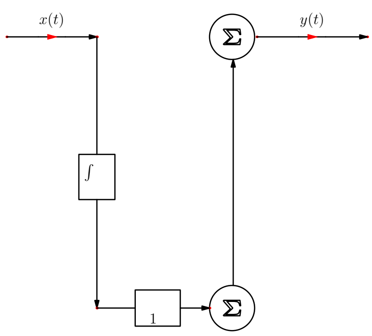
This can be an analogue signal which has been sampled, or a digital signal from a start. Before we have used differential equations to describe our system. This will not work for discrete time signals. Instead we have to use difference equations. For contionous LTI system block diagrams contained integrators, but what will be the equivalent for DSP?

Continous

$$\sum_{i=0}^{N} a_i \int_{(i)} y\mathrm{d}t = \sum_{j=0}^{N} b_j \int_{(j)} x\mathrm{d}t$$

Example:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(t)$$
$$y = \int x(t)\mathrm{d}t$$

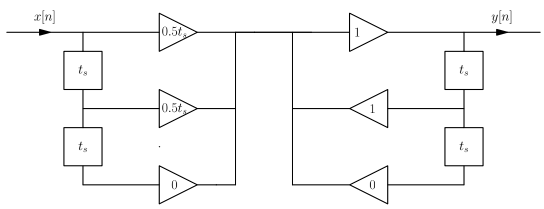


Discrete

$$\sum_{n=0}^{N} a_n y[k-n] = \sum_{n=0}^{N} b_n x[k-n]$$

Example:

$$\begin{aligned}
\frac{\mathrm{d}y}{\mathrm{d}t} &= x(t) \\
y &= \int x(t)\mathrm{d}t \\
y[n] &= y[n-1] + \frac{1}{2}t_s\left(x[n] + x[n-1]\right) \\
y[n] - y[n-1] &= \frac{1}{2}t_s x[n] + \frac{1}{2}t_s x[n-1]
\end{aligned}$$



So now integration and differentiation will be implemented through delay circuits!

Definition 26 The difference equation is a formula for computing an output sample at time n based on past and present input samples (x[n], x[n-1], x[n-2],..) and past output samples (y[n-1], y[n-2], ...) in the time domain.

$$\sum_{n=0}^{N} a_n y[k-n] = \sum_{n=0}^{N} b_n x[k-n] \tag{64}$$

We can define δ function as

$$\delta[k] = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \tag{65}$$

Then

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

Now integration corresponds to summation. Differentiation will correspond to difference between adjacent pulses.

Unit step impulse:

$$\epsilon[k] = \begin{cases} 1 & k \geq 0 \\ 0 & k < 0 \end{cases}$$

Consider a signal: x[k] = [2 $\underline{1}$ 1.5 1] for k = [-1 0 1 2]

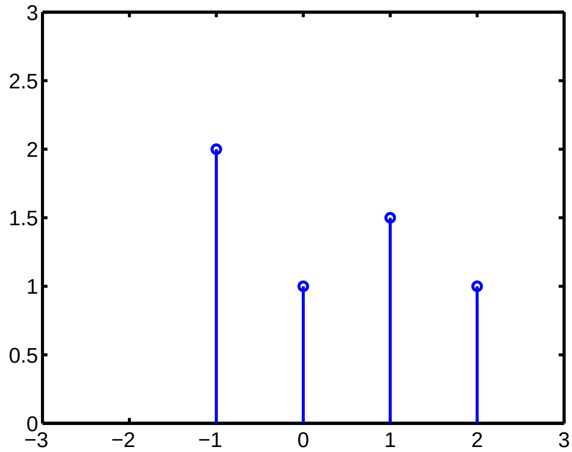


Figure 61: signal: x[k] = [2 $\underline{1}$ 1.5 1] defined for k = [-1 0 1 2]

This signal can be represented as:

$$x[n] = 2\delta[n+1] + \delta[n] + 1.5\delta[n-1] + 1\delta[n-2]$$
$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

Now $x[k]$ is the weight/amplitude. So what is the point? If we know the impulse response h[k] of a system, then we can easily determine the response to ANY signal:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$$

which corresponds to convolution sum.

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

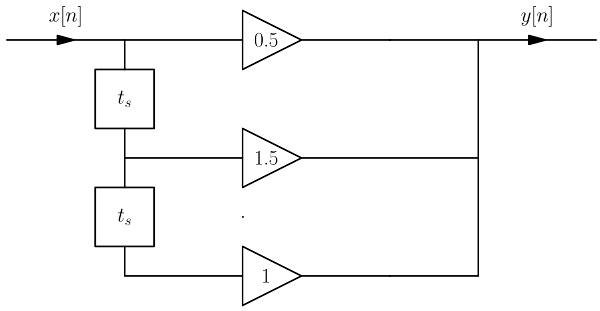


Figure 62:

◇ **Example 37.** Let our signal go through the digitized circuit:

$$y[n] = 0.5x[n] + 1.5x[n-1] + x[n-2]$$

We first find the impulse response ($\delta[0]$ as input):

$$\begin{aligned} h[0] &= 0.5\delta[0] + 1.5\delta[-1] + \delta[-2] = 0.5 \\ h[1] &= 0.5\delta[1] + 1.5\delta[0] + \delta[-1] = 1.5 \\ h[2] &= 0.5\delta[2] + 1.5\delta[1] + \delta[0] = 1 \end{aligned}$$

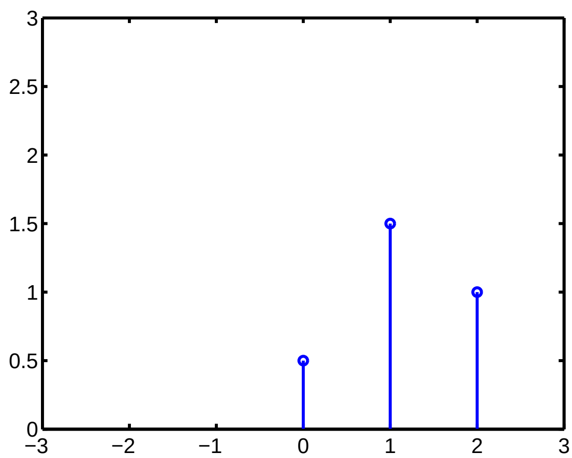


Figure 63:

Now, to get the output signal we need to calculate convolution between $x[n]$ and $h[n]$ where:

$$x[n] = \{2, \underline{1}, 1.5, 1\}$$
$$h[n] = \{\underline{0.5}, 1.5, 1\}$$

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$$

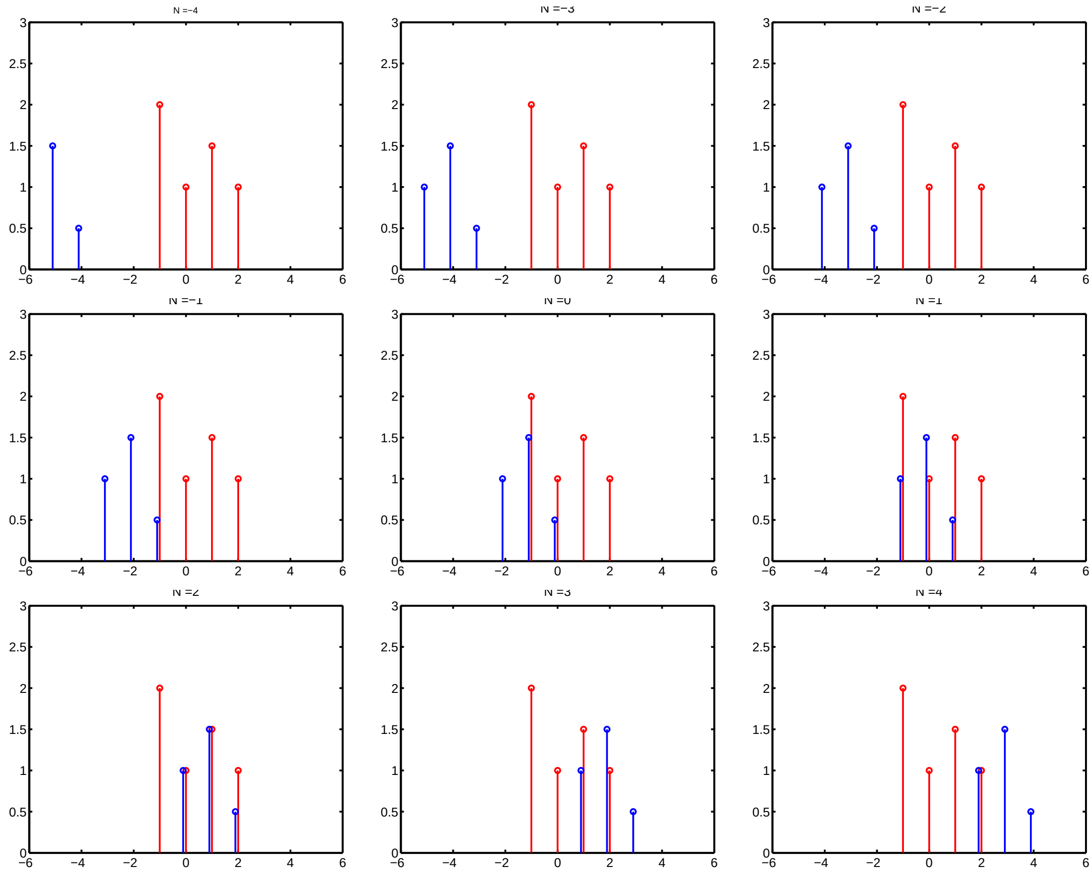


Figure 64: Convolution between $x[k]$ and $h[n-k]$

From the figure 64 we get:

$$y[n] \;=\; \{1, \underline{3.5}, 4.25, 3.75, 3, 1\}$$

We can get the same from difference equation:

$$\begin{aligned} y[n] &= 0.5x[n] + 1.5x[n-1] + x[n-2] \\ x[n] &= \{2, 1, 1.5, 1\} \end{aligned}$$

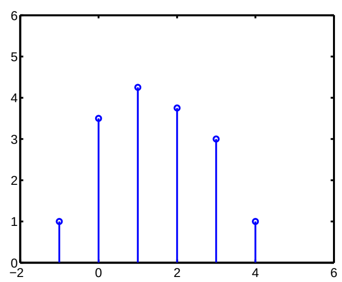


Figure 65: Convolution between $x[k]$ and $h[n-k]$

$$\begin{aligned}
y[-1] &= 0.5x[-1] + 1.5x[-1-1] + x[-1-2] = 1 + 0 + 0 = 1 \\
y[0] &= 0.5x[0] + 1.5x[-1] + x[-2] = 1 + 3 + 0 = 3.5 \\
y[1] &= 4.25 \\
y[2] &= 3.75 \\
y[3] &= 3 \\
y[4] &= 1 \\
y[n] &= \{1, \underline{3.5}, 4.25, 3.75, 3, 1\}
\end{aligned}$$

---

11 DTFT and DFT

11.1 Discrete Time Fourier Transform

Definition 27 Discrete Time Fourier Transform (DTFT) is defined as:

$$X(e^{j\omega}) = \mathcal{F}_* \{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \quad -\pi \le \omega < \pi \tag{66}$$

$$X(e^{j\omega}) = X(e^{j\omega+2\pi}) \tag{67}$$

ω is called "discrete fraquency" an it is not the same as the frequency for a continuous-time signal.

Inverse transform:

$$x[k] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{jk\omega} \mathrm{d}\omega$$

NOTE: This operation requires integration, which is not possible to implement for an arbitrary $X(e^{j\omega})$ on a computer.

Series in transformed into continuous function of real variable ω. DTFT measures the frequency content of a discrete-time signal. $X(e^{j\omega})$ is periodic and frequencies $[-\pi, \pi)$ need to be considered.

A sufficient condition to show the existence of the $\mathcal{F}_* \{\}$ is:

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty \tag{68}$$

$$\tag{69}$$

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s)$$

$$\mathcal{F}\{x_s(t)\} = \sum_{n=-\infty}^{\infty} x(nT_s)\mathcal{F}\{\delta(t - nT_s)\} = \sum_{n=-\infty}^{\infty} x(nT_s)\mathrm{e}^{-jn\Omega T_s}$$

If we now let $\omega = \Omega T_s$, then

$$X_s\left(e^{j\omega}\right) = X_s\left(e^{j\Omega T_s}\right) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

At the same time:

$$X_s(\Omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(\Omega - n\Omega_s) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X\left(\frac{\omega}{T_s} - \frac{2\pi n}{T_s}\right)$$

Which is periodic with a period of $2\pi/T_s$

11.2 Duality in Time and Frequency

In practice there are many signals of interest that does not satisfy the above absolute summability condition and therefore we can not find DTFT from the definition above. Duality in the time and frequency representations allows us to find DTFT of those signals. A DTFT of $\delta[n-k]$ for k being an integer is

$$\mathcal{F}_* \{\delta[n-k]\} = e^{-j\omega k}$$

and

$$\mathcal{F}_* \left\{e^{-j\omega_0 n}\right\} = 2\pi\delta(\omega+\omega_0) \quad -\pi \le \omega_0 < \pi$$

We can show that by calculating inverse DTFT of $2\pi\delta(\omega+\omega_0)$

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} 2\pi\delta(\omega+\omega_0)e^{jk\omega}\mathrm{d}\omega = e^{-j\omega k}\Big|_{\omega=\omega_0} = e^{-j\omega_0 k}$$

More, for

$$\begin{aligned} x[n] = A\delta[n] \quad &\Leftrightarrow \quad X(e^{j\omega}) = A \\ y[n] = A \quad -\infty < n < \infty \quad &\Leftrightarrow \quad Y(e^{j\omega}) = 2\pi A\delta(\omega) \quad -\pi \le \omega < \pi \end{aligned}$$

Using above we can calculate DTFT of a cos function:

$$\begin{aligned} x[n] &= \frac{1}{2}\left[e^{j(\omega_0 n+\theta)} + e^{-j(\omega_0 n+\theta)}\right] \\ X(e^{j\omega}) &= \pi\left[e^{j\theta}\delta(\omega-\omega_0) + e^{-j\theta}\delta(\omega+\omega_0)\right] \end{aligned}$$

11.3 Time and frequenct shift

$$\begin{aligned} x[n-N] \quad &\Leftrightarrow \quad X(e^{j\omega})\cdot e^{-j\omega N} \\ x[n]\, e^{j\omega_0 n} \quad &\Leftrightarrow \quad X(e^{j\omega - j\omega_0}) \end{aligned}$$

where $x[n]\, e^{i\omega_0 n}$ is modulated $x[n]$ because it is multiplied by complex discrete time sinusoids

$$x[n]\, e^{j\omega_0 n} = x[n]\cos(\omega_0 n) + jx[n]\sin(\omega_0 n)$$

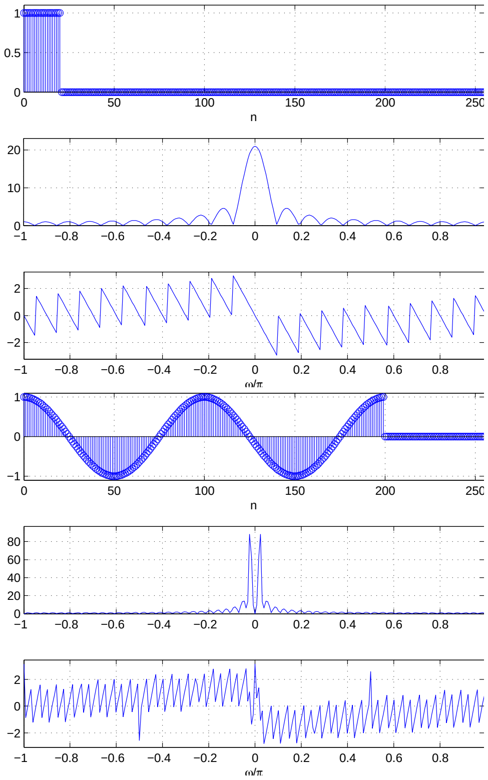
11.4 Computing DTFT using Matlab

ω is related to the sampling frequency by:

$$\Omega = \frac{\omega}{T_s}$$

Since $\omega \in [-\pi, \pi)$ corresponds to $\Omega \in [-\pi/T_s, \pi/T_s)$ or $\Omega \in [-\Omega_s/2, \Omega_s/2)$
Figure 10.2

11.5 Discrete Fourier Transform (DFT)




```
 %
% DTFT of aperiodic signals
% (pulse, windowed sinusoid, chirp)
%
clear all; clf
L=256; % length of signal (including zeros)

% signals
ind=input(' Pulse (1) or windowed sinusoid (2) or chirp (3)  ?  ')
if ind==1,
N=21;x=[ones(1,N) zeros(1,L-N)];                          % pulse
elseif ind==2,
N=200; n=0:N-1;x=[cos(4*pi*n/N) zeros(1,L-N)];   % windowed sinusoid
else
n=0:L-1;x=cos(pi*n.^2/(4*L));                                  % chirp
end

% DTFT
X=fft(x);
w=0:2*pi/L:2*pi-2*pi/L;w1=(w-pi)/pi; % frequencies
n=0:length(x)-1;
h = figure(1);
set(h,'PaperUnits','points');
set(h,'PaperSize',[400 320]);
set(h,'PaperPositionMode','auto')

subplot(311)
stem(n,x); axis([0 length(n)-1 1.1*min(x) 1.1*max(x)]); grid
xlabel('n'); ylabel('x[n]')
subplot(312)
plot(w1,fftshift(abs(X)));axis([min(w1) max(w1) 0 1.1*max(abs(X))])
ylabel('|X(e^{j\omega})|'); grid
subplot(313)
plot(w1,fftshift(angle(X)))
axis([min(w1) max(w1) 1.1*min(angle(X)) 1.1*max(angle(X))])
ylabel('<X(e^{j\omega})'); xlabel('\omega/\pi'); grid
print('-dpdf','-r300','DTFT1')
```

Figure 66:

Definition 28 Given a periodic signal $x[n]$ of period N, we can define its **Discrete Fourier Transform DFT** as:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi nk/N} \quad 0 \le k \le N-1 \tag{70}$$

And inverse DFT

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j2\pi nk/N} \quad 0 \le n < N \tag{71}$$

$$\omega_k = 2\pi k/N \tag{72}$$

Here both $x[n]$ and $X[n]$ are periodic of the same period N. When we use DFT to calculate FT of a sampled signal, we should remember that sampling of the input signal has two consequences. First, it will make FT periodic, which is ok as DFT calculates only one period. But also the transform of a periodic signal is mulitplied by a factor T_s^{-1}. So to make DFT calculated transform match the analytical expression (as in Matlab 2 exercise), one need to multiply the transform form DTF by T_s.

11.6 DFT of Aperiodic Discrete-Time Signals

For an aperiodic discrete-time signals, DFT can be calculate by sampling of the DTFT in the frequency space. Suppose we choose $\omega_k = 2\pi k/L$ as a sampling frequency, where the appropriate value for integer L need to be determined. Analogues with the sampling in time domain we have discussed before, sampling in frequency domain generates a periodic signal in time!

$$\tilde{x}[n] = \sum_{r=-\infty}^{\infty} x[n + rL]$$

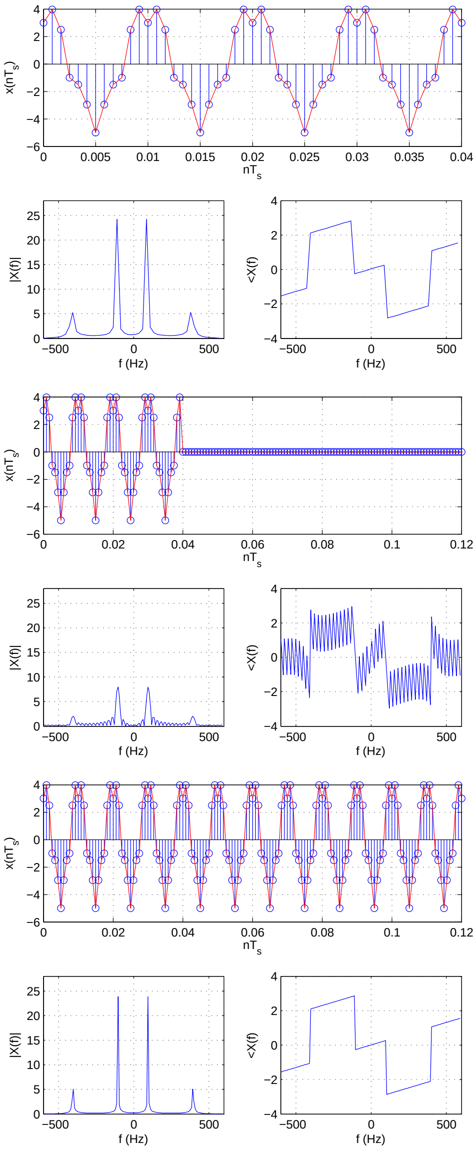
Now if $x[n]$ is of finite length N, then when $L \geq N$ the periodic expansion $\tilde{x}[n]$ clearly displays the first period equal to $x[n]$ with some zeros at the end when $L > N$. On the other hand when $L < N$ then: $\tilde{x}[n] \neq x[n]$ because of aliasing in the time domain.

$$\begin{aligned}
\tilde{x}[n] &= \frac{1}{L}\sum_{k=0}^{L-1} \tilde{X}[k]\, e^{j2\pi nk/L} \quad 0 \leq n \leq L-1 \\
\tilde{X}[k] &= \sum_{n=0}^{L-1} \tilde{x}[n]\, e^{-j2\pi nk/L} \quad 0 \leq k \leq L-1
\end{aligned}$$

Then:

- $X[k] = \tilde{X}[k]$ for $0 \leq k \leq L-1$ is the DFT of $x[n]$
- $x[n] = \tilde{x}[n]\, W[n]$ where $W[n] = u[n] - u[n-L]$ is a rectangular window of length N, is the IDFT if $X[k]$. The IDFT $x[n]$ is defined for $0 \leq n \leq L-1$

For a periodic signal we can improve the resolution of DFT by including more periods. Here the peaks will appear at the same frequency, but they are sharper as the "nearest" zeros are separated by a smaller frequency difference $f_s = 1/T_s$. If we make L larger by including zeros, the DFT will include convolution with $\operatorname{sinc}(\omega)$. This is because, in that case our signal is a multiplication of x[n] and a step function, so in the frequency domain this will be convolution with $\operatorname{sinc}(\omega)$) as can be seen on Figure 102.




```
% Example 10.21
% Improving frequency resolution of FFT of periodic signals
f0=100; f1=4*f0; % frequency in Hz
Ts=1/(3*f1); % sampling period
t=0:Ts:4/f0; % time for 4 periods
y=4*cos(2*pi*f0*t)-cos(2*pi*f1*t); % sampled signal (4 periods)
M=length(y);
Y=fft(y,M); Y=fftshift(Y)/4; % fft using 4 periods, shifting and normalizing
t1=0:Ts:12/f0; % time for 12 periods
y2=4*cos(2*pi*f0*t1)-cos(2*pi*f1*t1); % sampled signal (12 periods)
y2(size(y,2):size(y1,2))=0;
Y2=fft(y2);Y2=fftshift(Y2)/12; % fft using 12 periods, shifting and normalizing
w=2*[0:M-1]./M-1; f=w/(2*Ts); % frequency scale
t1=0:Ts:12/f0; % time for 12 periods
y1=4*cos(2*pi*f0*t1)-cos(2*pi*f1*t1); % sampled signal (12 periods)
Y1=fft(y1);Y1=fftshift(Y1)/12; % fft using 12 periods, shifting and normalizing
w=2*[0:M-1]./M-1; f=w/(2*Ts); % frequency scale

h = figure(1);
set(h,'PaperUnits','points'); set(h,'PaperSize',[400 320]);
set(h,'PaperPositionMode','auto')
subplot(211); stem(t,y); grid;
hold on;
plot(t,y,'r')
hold off; xlabel('nT_s'); ylabel('x(nT_s)')
subplot(223)
plot(f,abs(Y));grid; xlabel('f (Hz)'); ylabel('|X(f)|'); axis([-600 600 0 28])
subplot(224)
plot(f,angle(Y));grid; xlabel('f (Hz)'); ylabel('<X(f)'); axis([-600 600 -4 4])

N=length(y1); w=2*[0:N-1]./N-1;f=w/(2*Ts);
print('-dpdf','-r300','FFT1')

h = figure(2); set(h,'PaperUnits','points');
set(h,'PaperSize',[400 320]); set(h,'PaperPositionMode','auto')

subplot(211); stem(t1,y1);grid; hold on; plot(t1,y1,'r');
hold off; xlabel('nT_s'); ylabel('x(nT_s)')
subplot(223)
plot(f,abs(Y1));grid; ; xlabel('f (Hz)'); ylabel('|X(f)|'); axis([-600 600 0 28])
subplot(224)
plot(f,angle(Y1));grid; xlabel('f (Hz)'); ylabel('<X(f)'); axis([-600 600 -4 4])
print('-dpdf','-r300','FFT2')
h = figure(3)
set(h,'PaperUnits','points'); set(h,'PaperSize',[400 320]);
set(h,'PaperPositionMode','auto'); subplot(211)
stem(t1,y2);grid; hold on
plot(t1,y2,'r'); hold off; xlabel('nT_s'); ylabel('x(nT_s)')
subplot(223)
plot(f,abs(Y2));grid; ; xlabel('f (Hz)'); ylabel('|X(f)|'); axis([-600 600 0 28])
subplot(224)
plot(f,angle(Y2));grid; xlabel('f (Hz)'); ylabel('<X(f)'); axis([-600 600 -4 4])
print('-dpdf','-r300','FFT3')
```

Figure 67: DFT

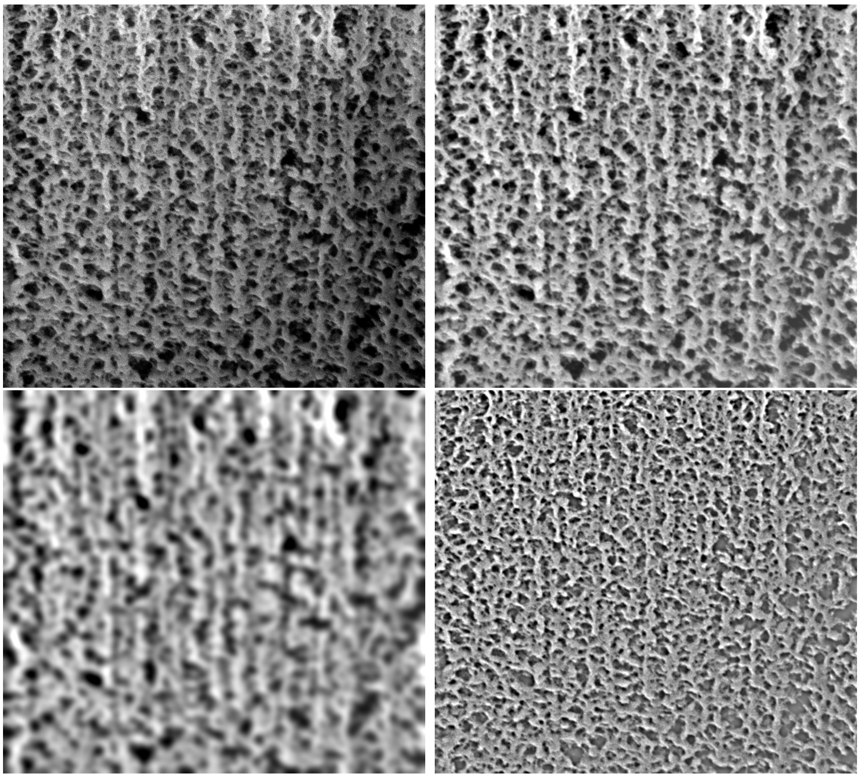


Figure 68: 2D FFT filtering of a microscopy image (done using ImageJ band bass filters): original image (size approx 1000x1000 pixels); image with structures smaller the 3pix and larger then 200 pix filtered out; image with structures smaller the 20pix and larger then 200 pix filtered out; image with structures smaller the 3pix and larger then 20 pix filtered out.

◇ **Example 38.** DTFT
Find DTFT and DFT of a discrete rect function.

$$x[n] = \begin{cases} 1 & 0 \leq n < M \\ 0 & \text{otherwise} \end{cases}$$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} = \sum_{n=0}^{M-1} x[n] e^{-j\omega n} = \sum_{n=0}^{M-1} e^{-j\omega n} =$$
$$= e^{-j\omega 0} + e^{-j\omega 1} + e^{-j\omega 2} + ... + e^{-j\omega(M-1)}$$

Using the property of a geometric series:

$$1 + r + r^2 + ... + r^{M-1} = \frac{1 - r^M}{1 - r}$$

Then,

$$X(e^{j\omega}) = e^{-j\omega 0} + e^{-j\omega 1} + e^{-j\omega 2} + ..e^{-j\omega(M-1)} = \frac{1 - e^{-j\omega M}}{1 - e^{-j\omega}} =$$
$$= \frac{e^{-j\omega M/2}}{e^{-j\omega/2}} \frac{e^{j\omega M/2} - e^{-j\omega M}}{e^{j\omega/2} - e^{-j\omega/2}} = e^{-j\omega(M-1)/2} \frac{\sin(\omega M/2)}{\sin(\omega/2)} =$$
$$= e^{-j\omega(M-1)/2} \frac{\sin(\omega M/2)}{\omega M/2} \frac{\omega/2}{\sin(\omega/2)} \frac{\omega M/2}{\omega/2} = e^{-j\omega(M-1)/2} \frac{\mathrm{sinc}(\omega \mathrm{M}/2)}{\mathrm{sinc}(\omega/2)} M$$

```
M=4; %rect length
N = 50; % signal size for DFT (by definition it is infnity for DTFT)
w=0:0.001*pi:2*pi   %

dtft=M.*sinc(w.*M./2./pi)./(sinc(w./2./pi)).*exp(-j.*w.*(M-1)./2);
subplot(2,2,1)
Mag=abs(dtft);
plot(w./pi*.5,Mag);title('Amplitude DTFT')

subplot(2,2,2)
Pha=angle(dtft);
plot(w./pi*.5,Pha);
title('Phase DTFT')


x = zeros(N,1);
wn = 0:2*pi/N:2*pi()*(N-1)/N;
x(1:M) = 1;
dftx = fft(x);
subplot(2,2,3)
Mag=abs(dftx);
stem(wn./pi*.5,Mag);
title('Amplitude DFT')
subplot(2,2,4)
Pha=angle(dftx);
stem(wn./pi*.5,Pha);
title('Phase DFT')
```

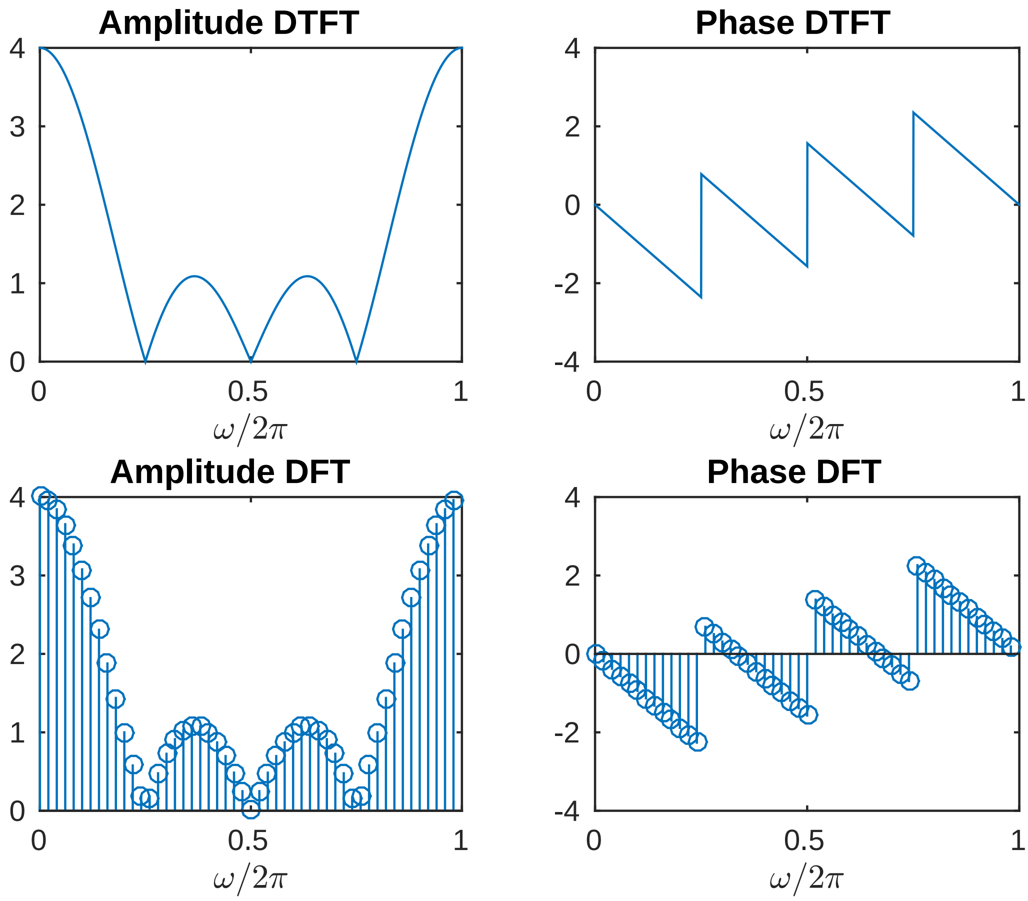


Figure 69: DTFT and DFT (matlab) calculated for a signal x[n] = 1 for $0 \leq n \leq 4$, as described in the example above. DFT can be calculated from analyticalformula for DTFT, by setting $\omega = 2\pi k/N$

◇ **Example 39.** DFT
Lets calculate DFT of the same signal using it's definition given above.

$$x[n] = \begin{cases} 1 & 0 \leq n < M \\ 0 & \text{otherwise} \end{cases}$$

For DFT the signal has to be periodic, with periodicity N, where $N > M$. For example:

$$\tilde{x}[n] = \begin{cases} 1 & 0 \leq n < M \\ 0 & M \leq n < N \end{cases}$$

For time continious periodic signal with period T_p (frequency $\omega_0 = 2\pi/T_p$), Fourier series expansion is given by:

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t} = \sum_{k=-\infty}^{\infty} a_k e^{j2\pi kt/T_p}$$

For discrete time signal, t $= nT_s$ and $T_p = Nt_s$

$$\tilde{x}[n] = \sum_{k=-\infty}^{\infty} a_k e^{j2\pi kn/N}$$

But there are only N unique frequencies in $\tilde{x}[n]$ due to periodicity of the complex exponentials.

$$\tilde{x}[n] = \sum_{k=0}^{N-1} a_k e^{j2\pi kn/N}$$

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] e^{j2\pi kn/N}$$

and we can show that

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j2\pi kn/N}$$

Back to our signal

$$\tilde{X}[k] = \sum_{n=0}^{M-1} e^{-j2\pi kn/N} = 1 + e^{-j2\pi k/N} + e^{-j2\pi k/N\cdot 2} + ... + e^{-j2\pi k/N\cdot(M-1)}$$

An this has the same form as for DTFT example above, but is only defined for discrete frequencies $0 \leq \omega = 2\pi k/N < 2\pi$ ♣