Lecture 5

Goals

- Be able to generate signals and noise in time and frequency domain in Matlab

Simulation and FFT Stuff

Consider a signal $x(t)$ with Fourier Transform $X(f)$. Suppose $X(f)$ is zero for $|f| > f_{max}$. Then according to the sampling theorem we can represent $x(t)$ by samples at rate $2f_{max}$. In addition assume that $x(t)$ is causal so that $x(t) = 0$ for $t < 0$. The Fourier transform of $x(t)$ is given by

$$\begin{aligned} X(f) &= \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt \\ &= \int_{0}^{\infty} x(t)e^{-j2\pi ft}dt \end{aligned}$$

Notice that $X(-f) = X^*(f)$. Similarly

$$\begin{aligned} x(t) &= \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}df \\ &= \int_{-f_{max}}^{f_{max}} X(f)e^{j2\pi ft}df \end{aligned}$$

Now we can approximate these integrals with finite sums. For the purpose of being able to use an FFT (Fast Fourier Transform) algorithm in MATLAB we let

$$\Delta f \Delta t = \frac{1}{N}.$$

Since $\Delta t = 1/(2f_{max})$, $\Delta f = 2f_{max}/N$. That is we are dividing the interval of the frequency response from $-f_{max}$ to f_{max} into N discrete frequencies. Then for $|f| < f_{max}$

$$\begin{aligned} X(f) &\approx \sum_{l=0}^{N-1} x(l\Delta t) e^{-j2\pi f l \Delta t} \Delta t \\ X_m = X(m\Delta f) &= \sum_{l=0}^{N-1} x(l\Delta t) \Delta t e^{-j2\pi m \Delta f l \Delta t} \\ &= \Delta t \sum_{l=0}^{N-1} x(l\Delta t) e^{-j2\pi m l/N}. \end{aligned}$$

Thus the above expression for X_m is valid for $0 \leq |m| \leq N/2$. Notice that $X_{-m} = X_{N-m}$. In MATLAB the operation FFT takes an input vector (indexed from 1 to N) and produces an output vector (indexed from 1 to N). The input-output relation is

$$X_m = \sum_{l=1}^{N} x_l e^{-j2\pi(m-1)(l-1)/N} \quad m = 1, \ldots, N$$

The input should correspond to the samples starting at time 0 up to time $(N-1)\Delta t$. Notice that in order to get the correct magnitude of the frequency response we must multiply what MATLAB produces by Δt. The output samples from 1 to $N/2+1$ (when normalized appropriately) correspond to the frequency response $X(0), X(\Delta f), \ldots, X(N\Delta f/2) = X(f_{max})$. The samples from $N/2+2$ to N correspond to the frequency response from $X(-f_{max}+\Delta f), \ldots X(-\Delta f)$. The order can be reversed into the logical order with the MATLAB command fftshift.

Inverse FFT

To perform an inverse FFT in MATLAB the spectrum should be rearranged so that the zero frequency shift corresponds to frequency sample 1, the maximum positive frequency ($f_{max} = N/2\Delta f$) corresponds to sample $N/2+1$, the maximum negative frequency $(-N/2+1)\Delta f$ corresponds to sample $N/2+2$ and the sample N corresponds to frequency $-\Delta f$.

$$
\begin{aligned}
x(t) &= \int_{-f_{max}}^{f_{max}} X(f) e^{j2\pi f t} df \\
x(t) &\approx \sum_{m=-N/2+1}^{N/2} X(m\Delta f) e^{j2\pi m\Delta f t} \Delta f \\
&\approx \sum_{m=0}^{N/2} X(m\Delta f) e^{j2\pi m\Delta f t} \Delta f + \sum_{m=N/2+1}^{N-1} X((m-N)\Delta f) e^{j2\pi (m-N)\Delta f t} \Delta f
\end{aligned}
$$

$$x(l\Delta t) \approx \sum_{m=0}^{N/2} X(m\Delta f)e^{j2\pi ml/N}\Delta f + \sum_{m=N/2+1}^{N-1} X((m-N)\Delta f)e^{j2\pi(m-N)l/N}\Delta f$$

Now if we let

$$X_m = \begin{cases} X(m\Delta f) & m = 0,1,...,N/2 \\ X((m-N)\Delta f) & m = N/2+1,...,N-1 \end{cases}$$

then we get

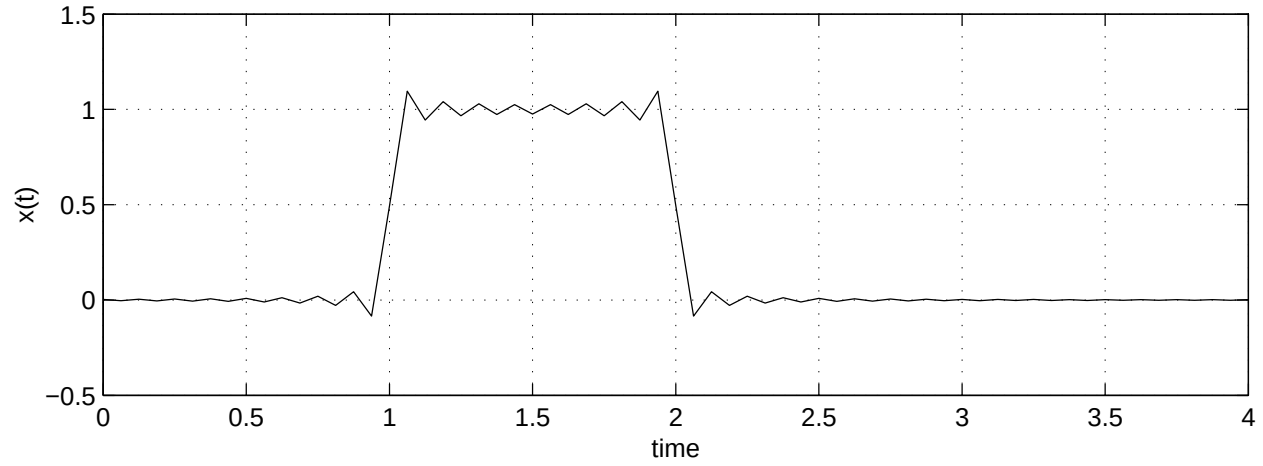
$$\begin{aligned} x(l\Delta t) &\approx \sum_{m=0}^{N-1} X_m e^{j2\pi ml/N}\Delta f \\ &\approx \frac{1}{\Delta t}\left[\frac{1}{N}\sum_{m=0}^{N-1} X_m e^{j2\pi ml/N}\right], \; l = 0,1,...,N-1. \end{aligned}$$

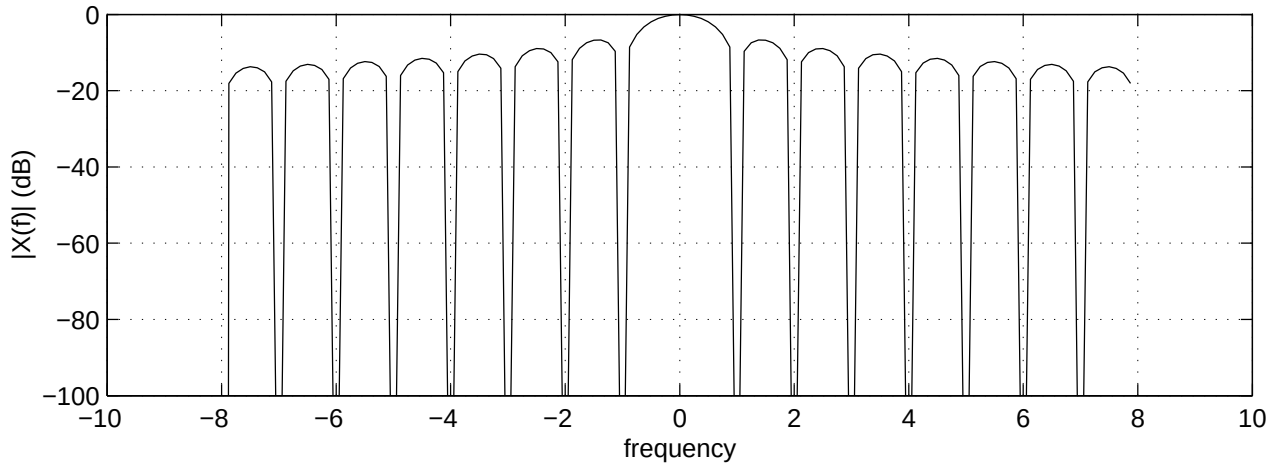
The MATLAB command ifft computes the expression

$$x_l = \frac{1}{N}\sum_{m=1}^{N} X_m e^{j2\pi(m-1)(l-1)/N} \quad l = 1,...,N$$

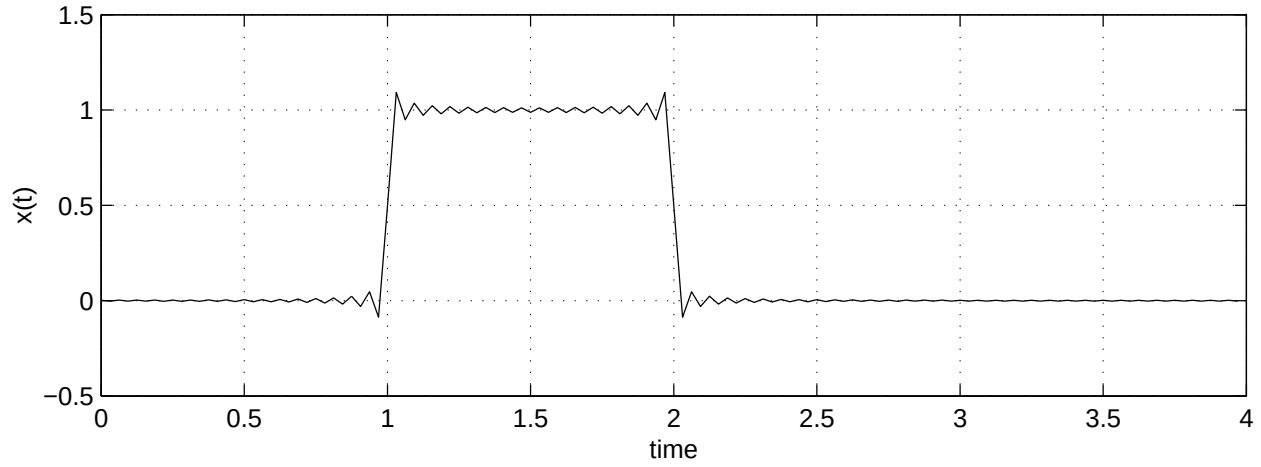
The time response goes from 0 to $(N-1)\Delta t = (N-1)/2f_{max}$ corresponds to the samples $x(0),\ldots,x((N-1)/(2f_{max}))$. If $X(-f) = X^*(f)$ then it is easy to show the the resulting signal after taking the ifft approximation will be real (as it should). However, because of numerical errors in computation typically when the inverse FFT is taken the imaginary part does not come out to be zero exactly. In this case we usually take just the real part of the answer.

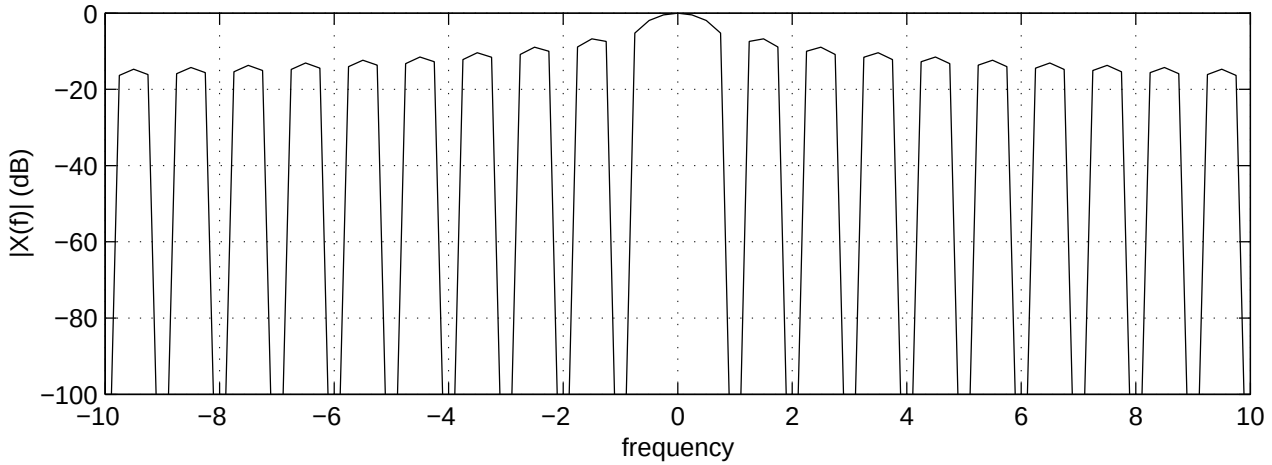
Example: Rectangular Pulse $f_{max} = 8,\ N = 128$



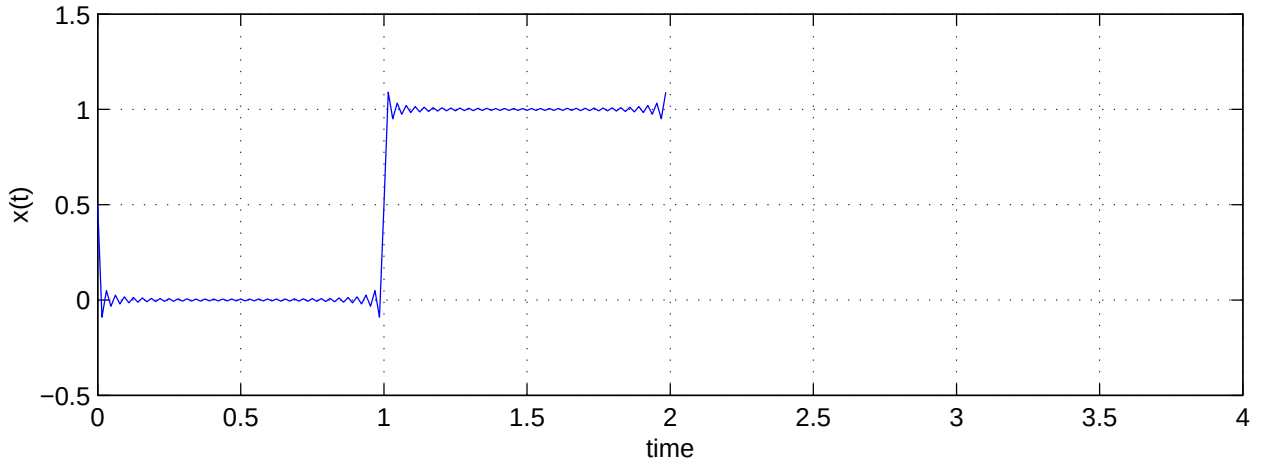


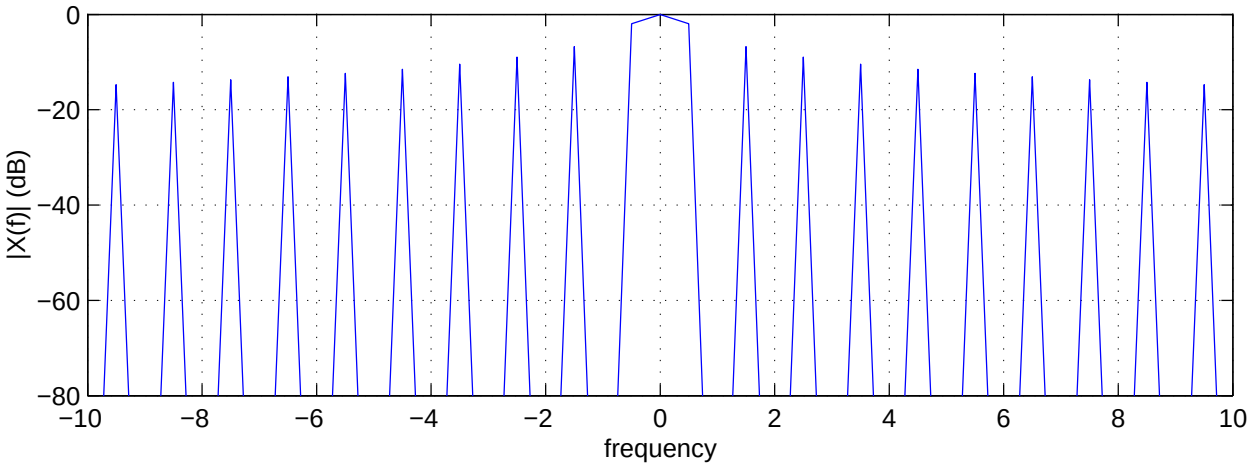
Example: Rectangular Pulse $f_{max} = 16,\ N = 128$



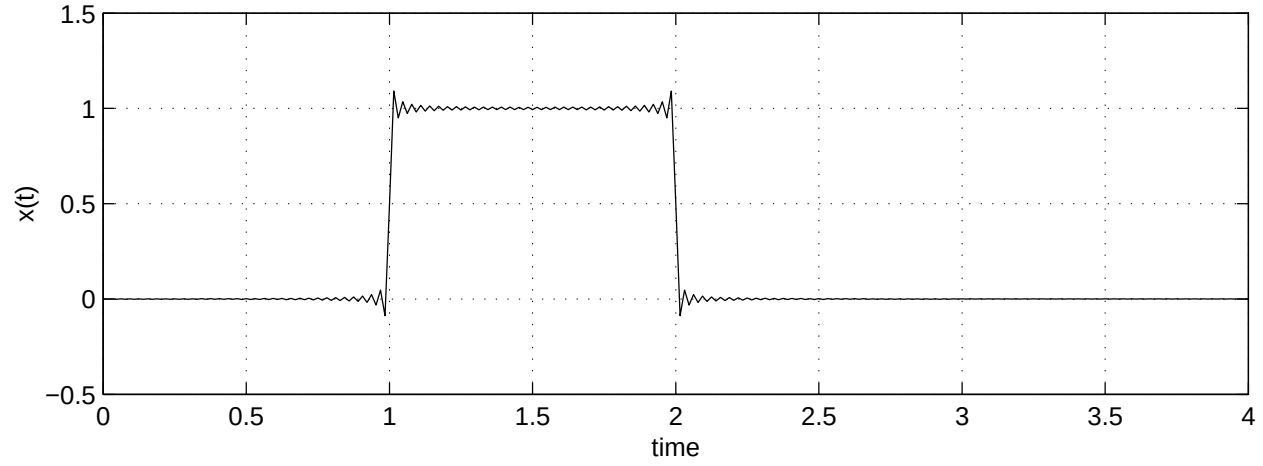


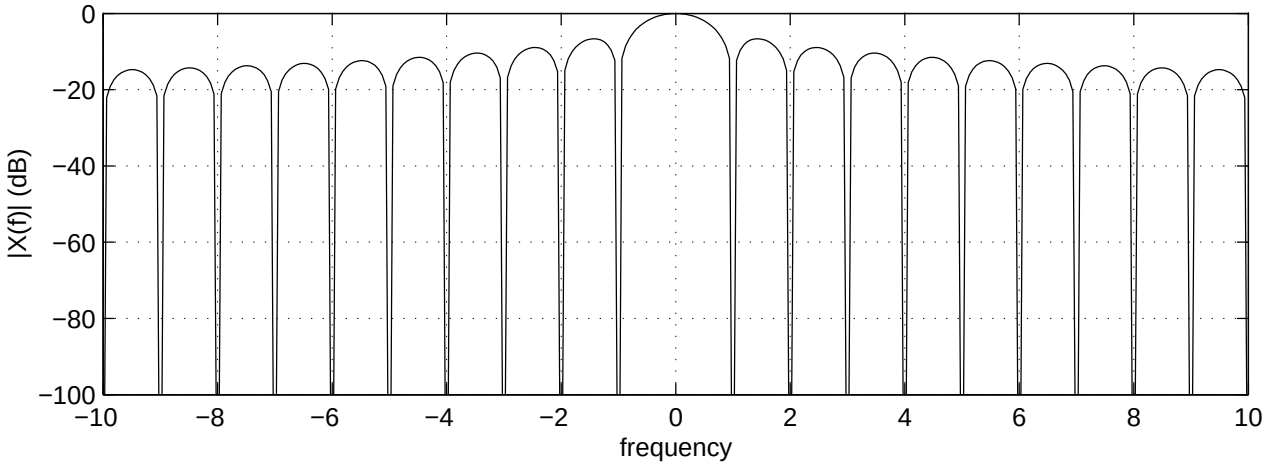
Example: Rectangular Pulse $f_{max} = 32,\ N = 128$



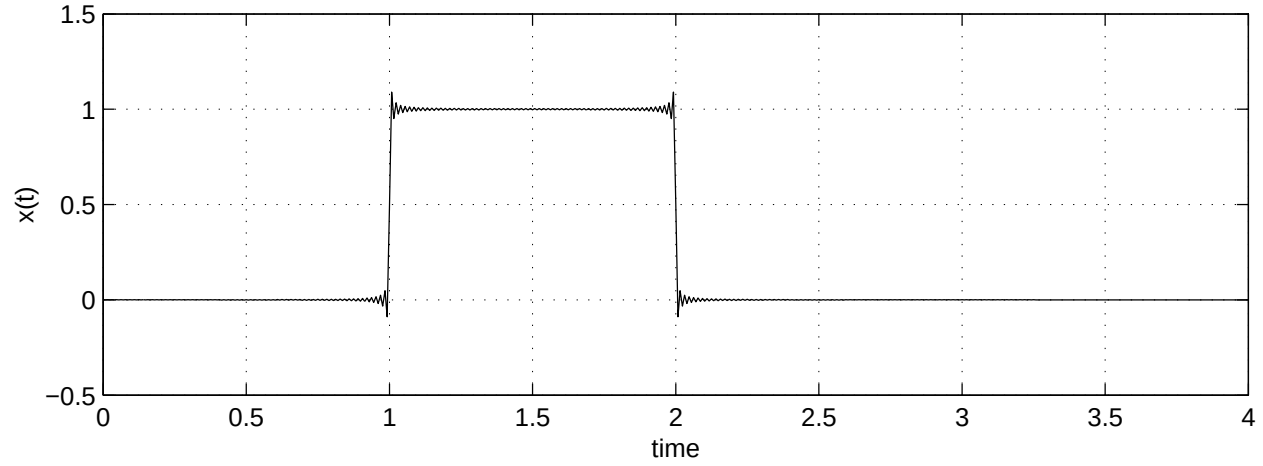


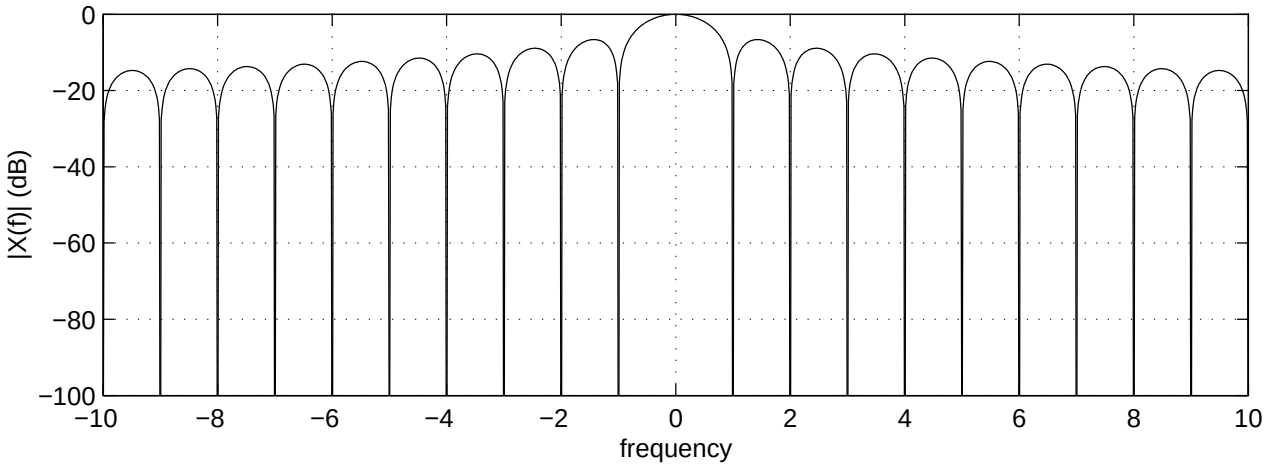
Example: Rectangular Pulse $f_{max} = 32,\ N = 1024$





Example: Rectangular Pulse $f_{max} = 64, N = 8192$





Simulating Noise

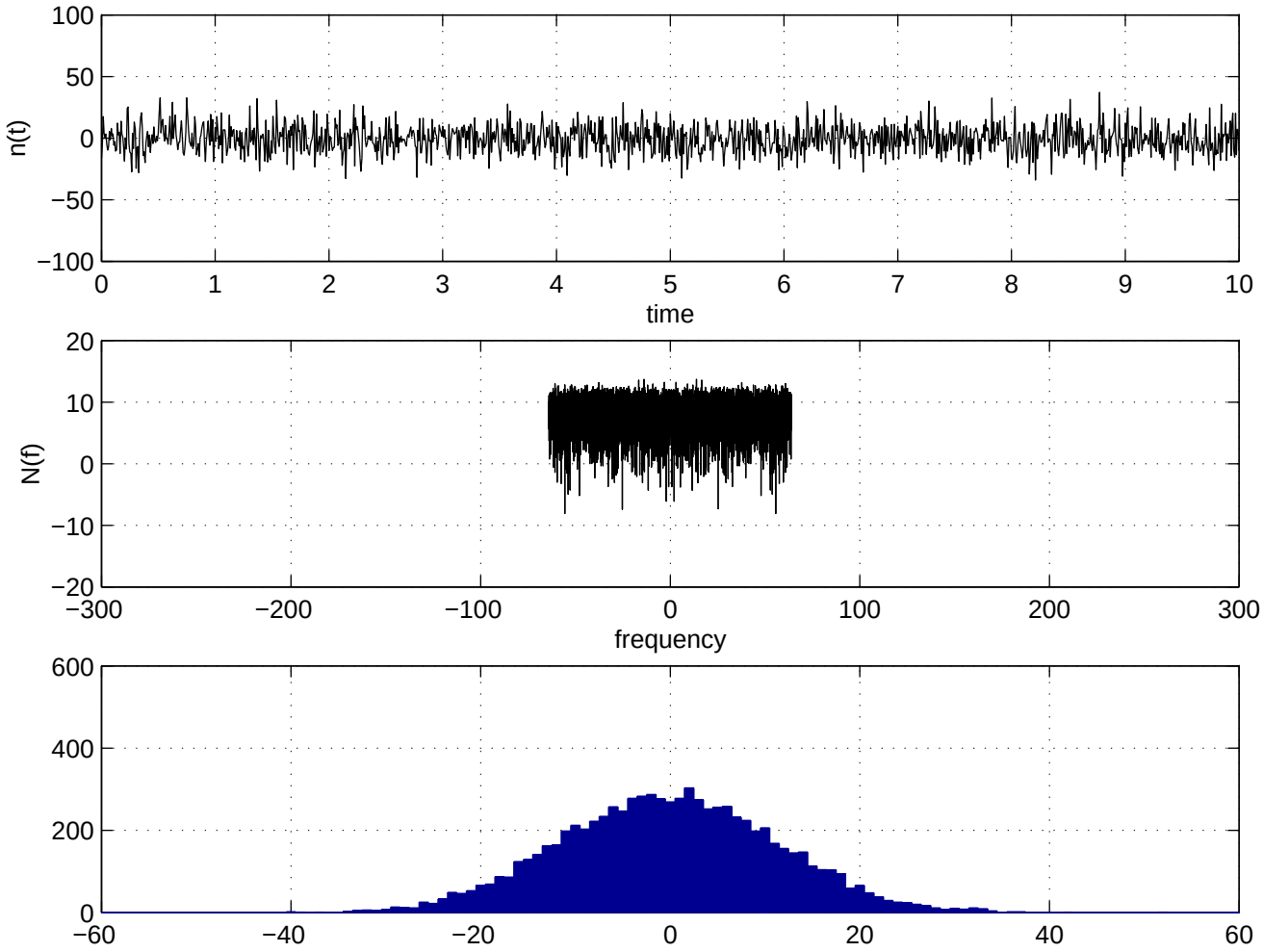
In a computer simulation of a communication system we represent a continuous time signal with samples according to the sampling theorem. A signal can be represented if the highest frequency content is less than half the sampling rate. Thus if f_s represents the sampling rate the simulation bandwidth is $f_s/2$. If we consider white Gaussian noise the noise has equal power at all frequencies up to the simulation bandwidth. Thus the power spectral density of the noise in the simulation is

$$S_n(f) = N_0/2, \; -f_s/2 < f < f_s/2$$

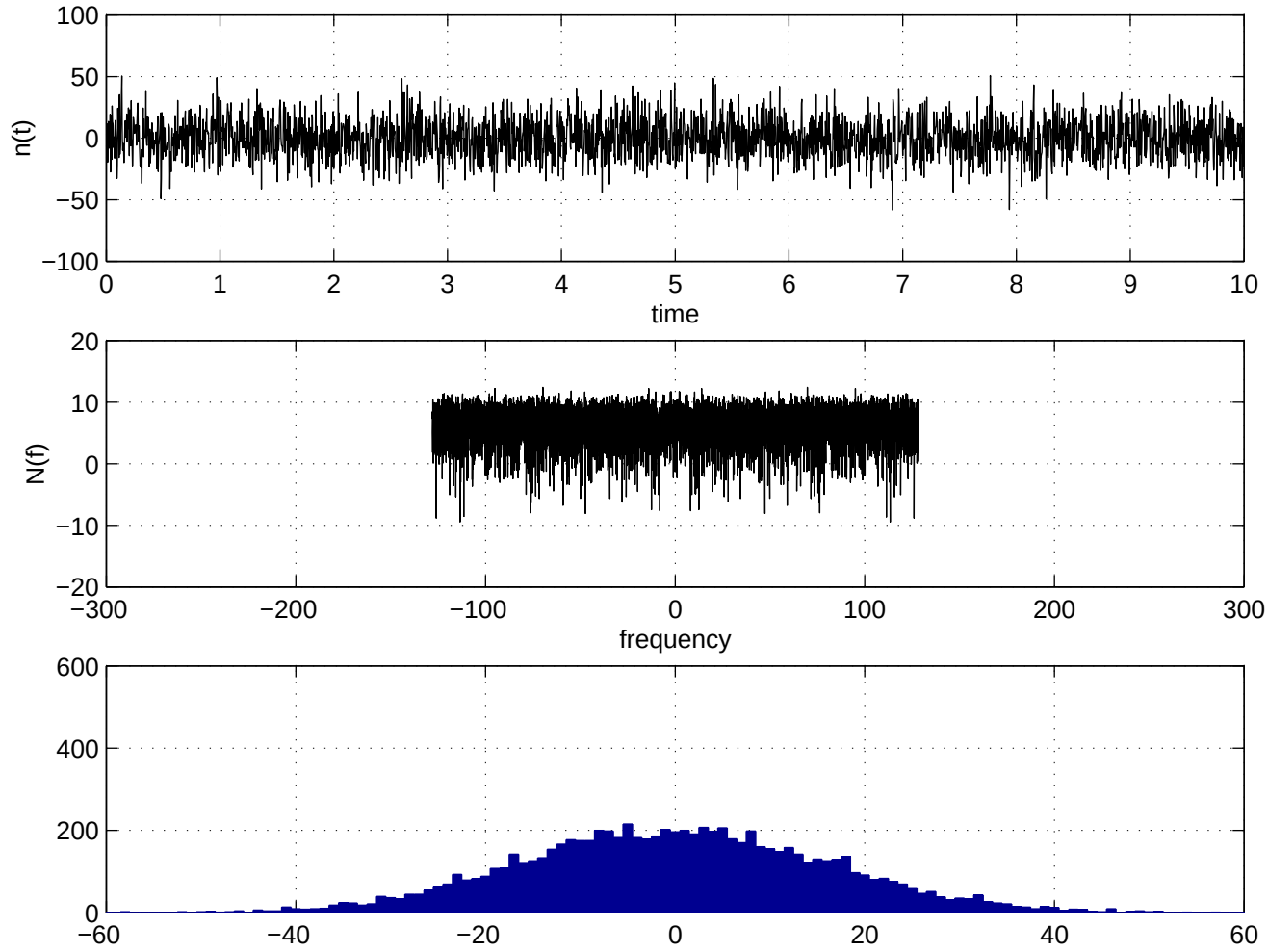
The noise samples (with this finite bandwidth) have variance

$$\sigma^2 = \int_{-f_s/2}^{f_s/2} S_n(f)\,df = \int_{-f_s/2}^{f_s/2} \frac{N_0}{2}\,df = N_0 f_{max}$$

Example: Noise $f_{max} = 64,\ N = 8192$



Example: Noise $f_{max} = 128,\ N = 8192$



Example: Noise $f_{max} = 256,\ N = 8192$