



Spectral theory of signal simulation and AI

Ivan Deykin

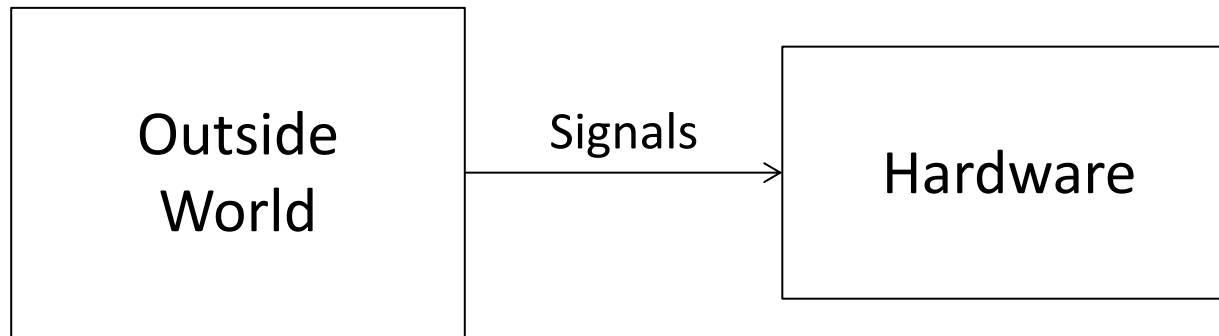
ideykin@mail.ru

Special thanks to Vladimir Syuzev and Elena Smirnova

IU6 (IC6) department, BMSTU

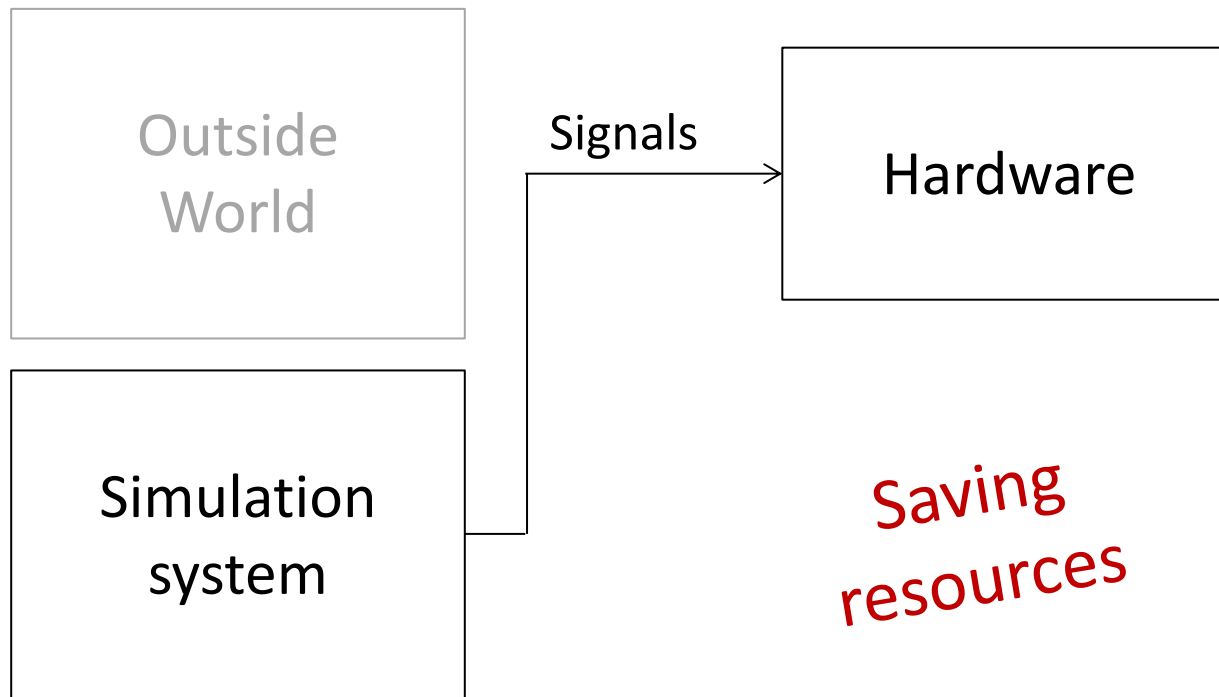
Signal simulation

Hardware-in-the-loop



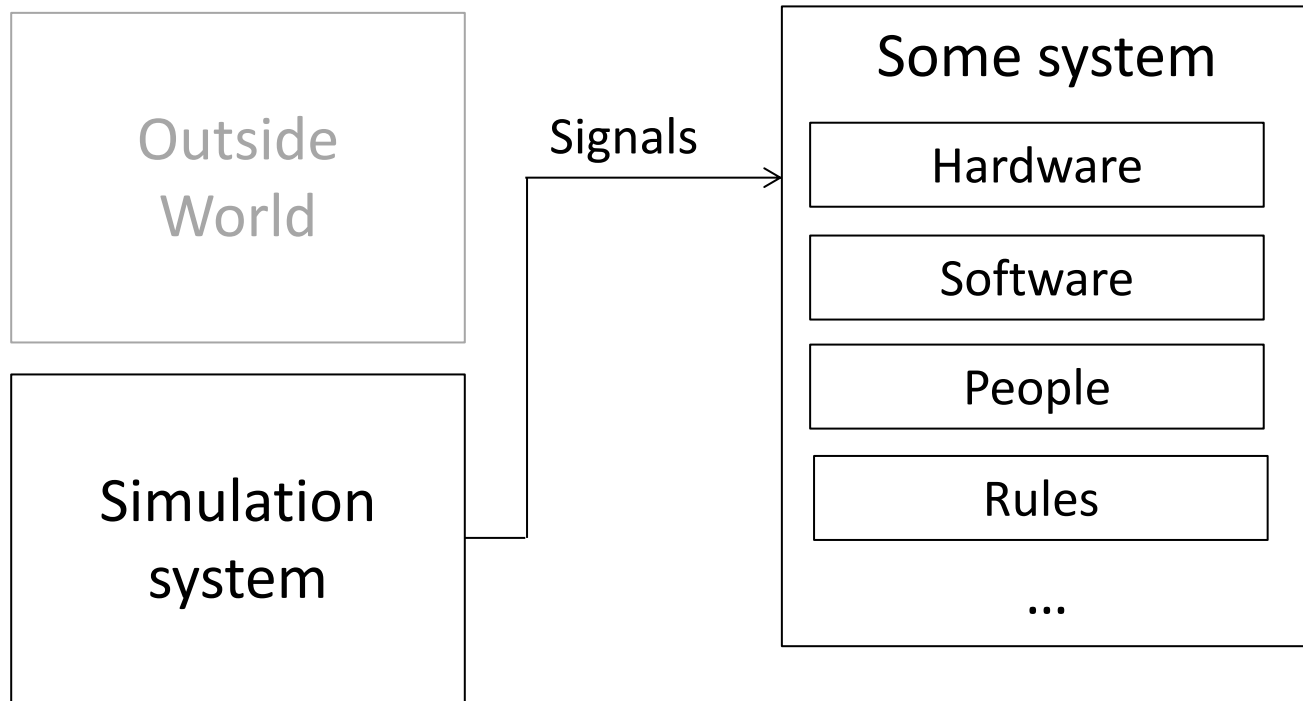
Signal simulation

Hardware-in-the-loop



Signal simulation

Hardware-in-the-loop

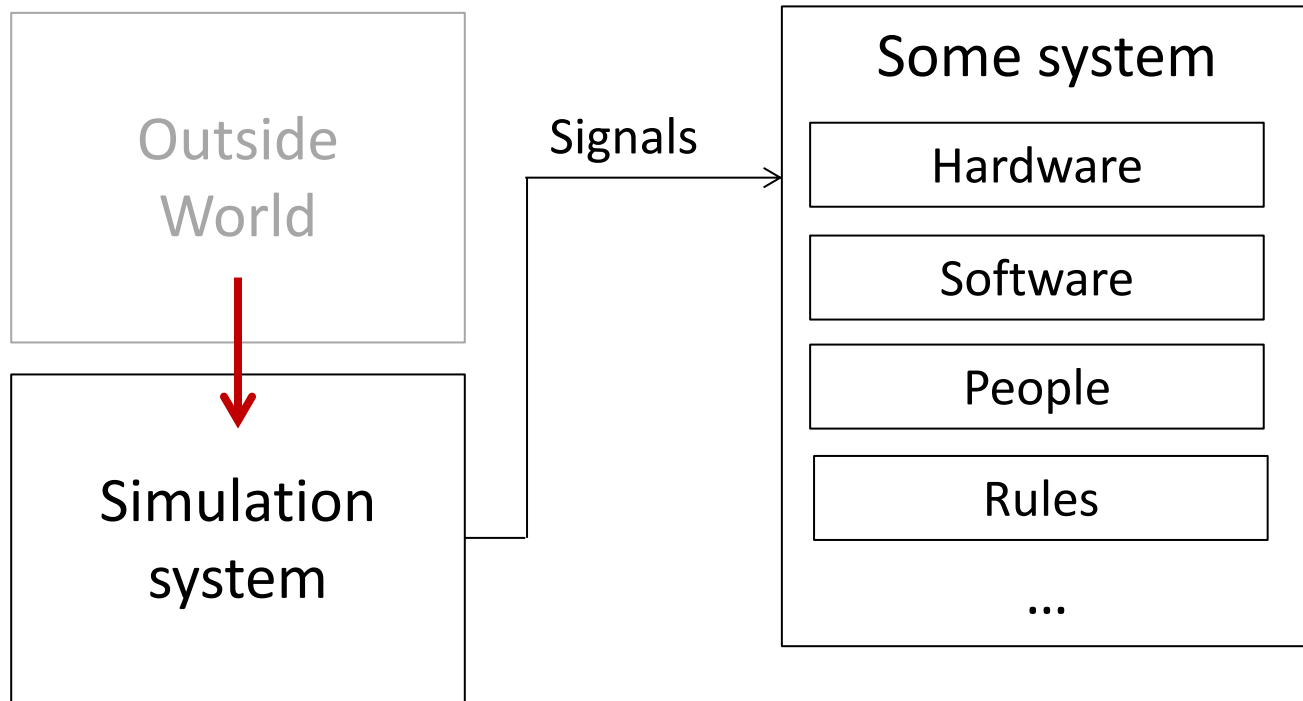


Example: forest fire prevention system

- Does hardware work correctly?
- Is software OK?
- Is staff properly trained?
- Are the rules adequate?

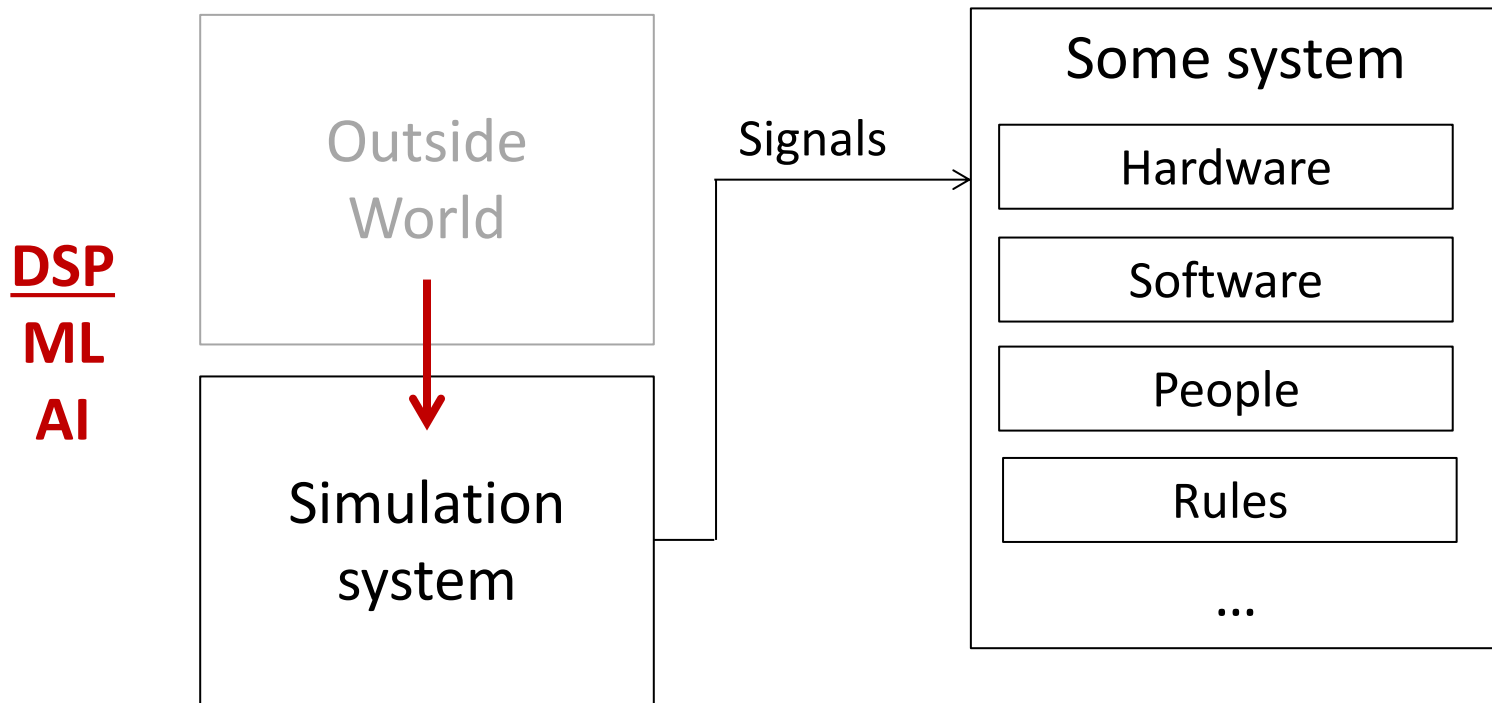
Signal simulation

Hardware-in-the-loop



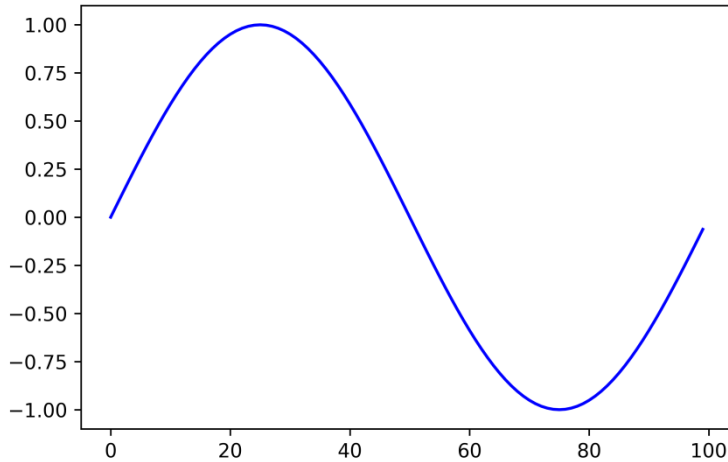
Signal simulation

Hardware-in-the-loop

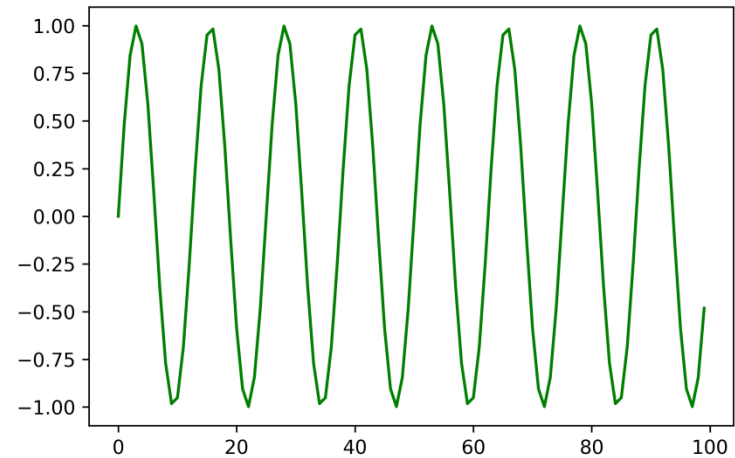


1D signal

$$x(t) = \sin(t) + \sin(8t); T = 2\pi, N = 100.$$

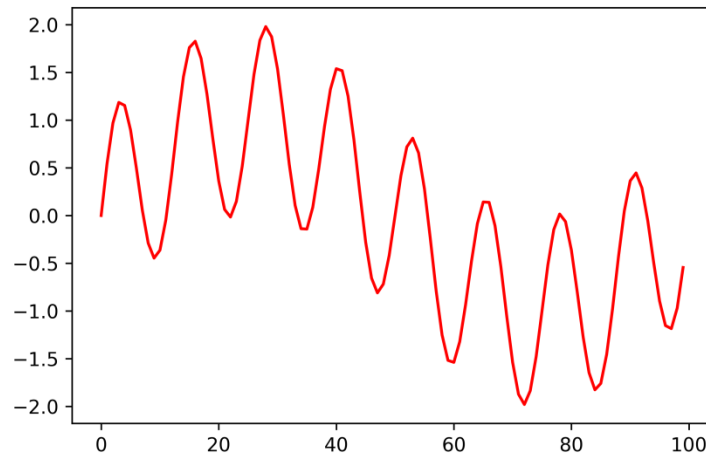


+



=

=



Time and frequency

$$f = \frac{n}{t} = \frac{1}{T}.$$

f – frequency – “the number of occurrences of a repeating event per unit of time”;

n – repeats of something within T ;

T – period.

Fourier Transform

Fourier – from time domain to frequency domain:

$$\mathcal{F}(x(t)) = X(f) = \int_{-\infty}^{+\infty} x(t)e^{-2\pi jft} dt,$$

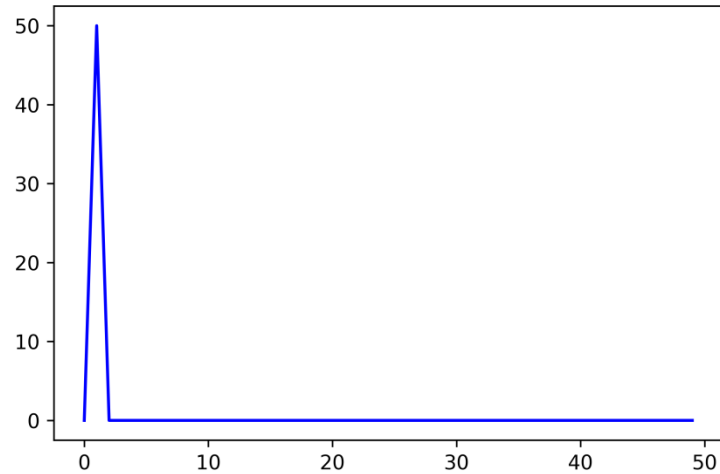
Inverse Fourier – from frequency to time:

$$\mathcal{F}^{-1}(X(f)) = x(t) = \int_{-\infty}^{+\infty} X(f)e^{2\pi jft} df.$$

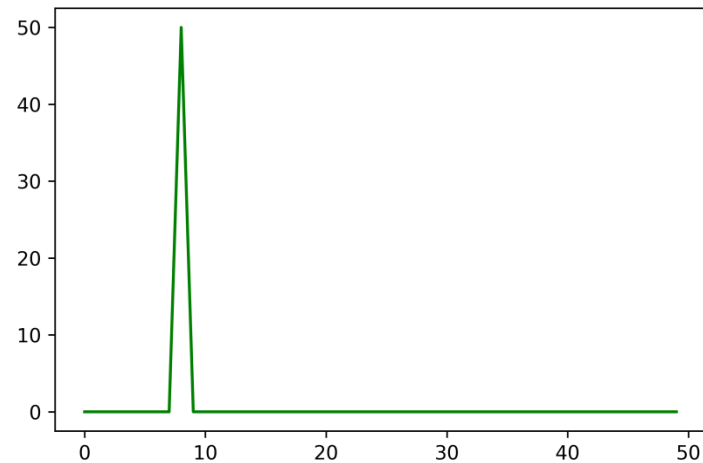
$$\omega = 2\pi f.$$

Fourier Transform

$$\mathcal{F}(\sin(t)) =$$

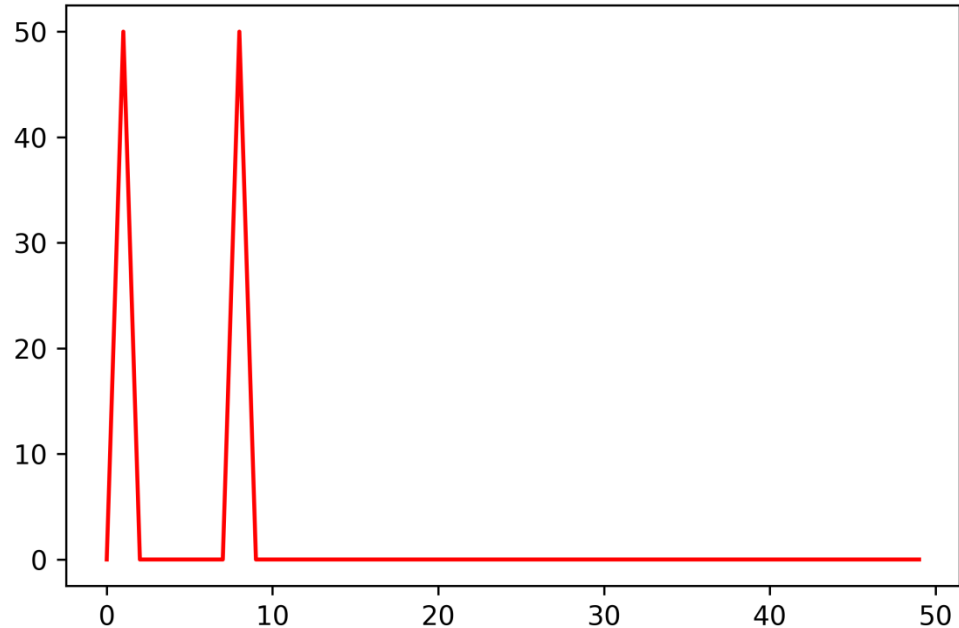


$$\mathcal{F}(\sin(8t)) =$$



Fourier Transform

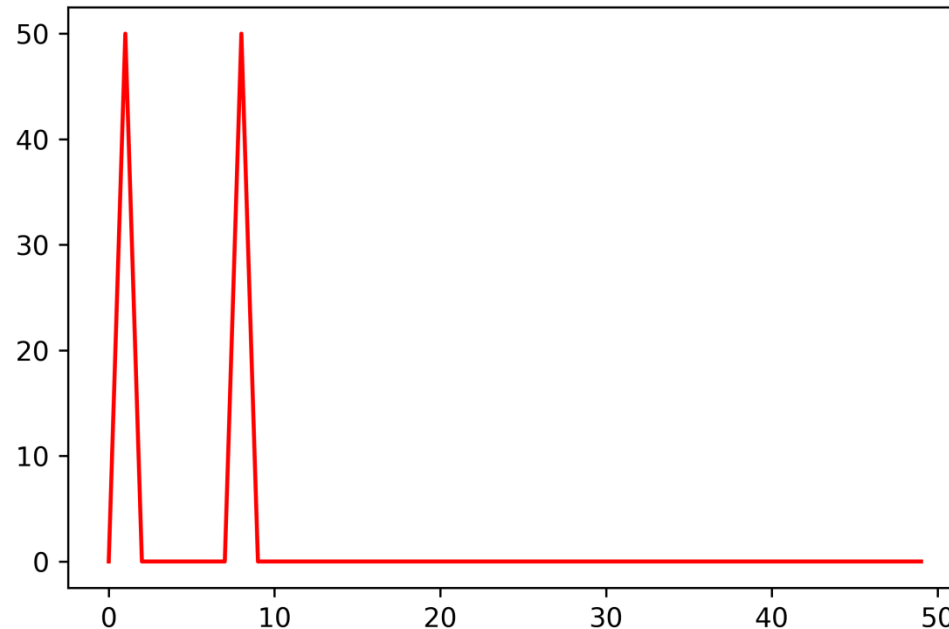
$$\mathcal{F}(x(t)) =$$



$\mathcal{F}(x(t))$ shows how “strong” or “loud” different frequencies are within a signal.

Fourier Transform

$$\mathcal{F}(x(t)) =$$

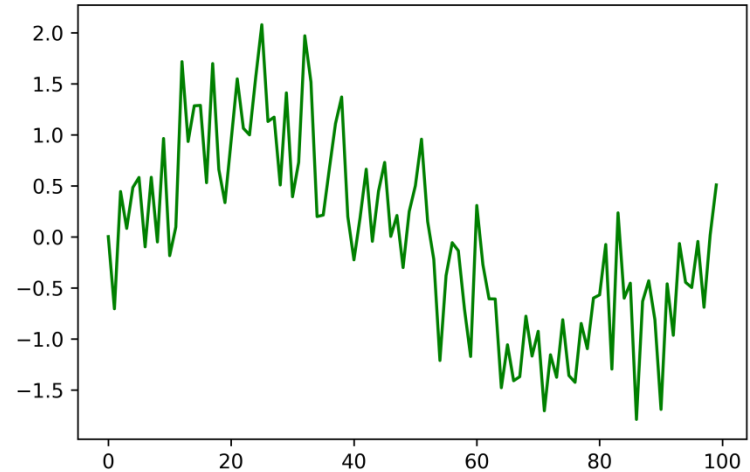
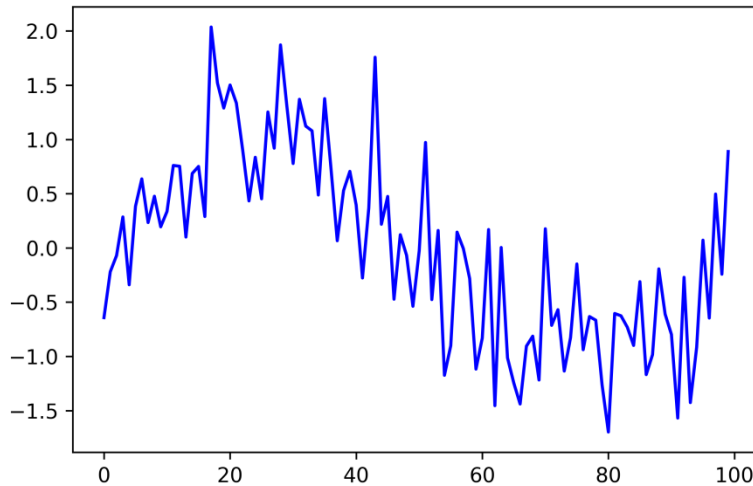


$\mathcal{F}(x(t)) = \mathcal{F}(\sin(t)) + \mathcal{F}(\sin(8t))$ - we can filter $\mathcal{F}(x(t))$ and use $\mathcal{F}^{-1}(X'(f))$ to reproduce the new signal

Random signal

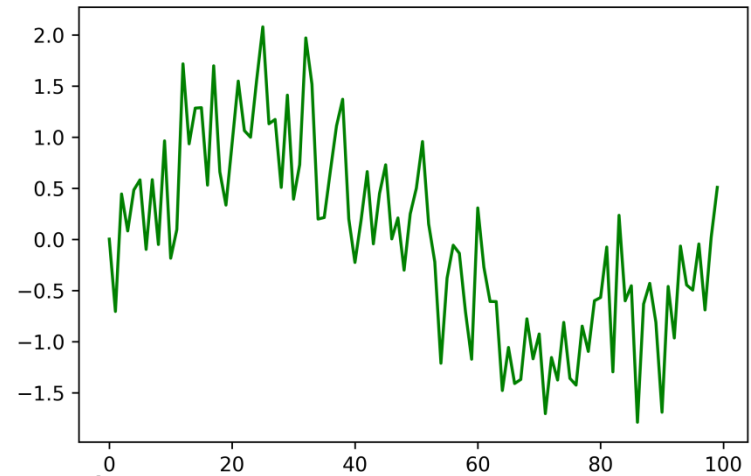
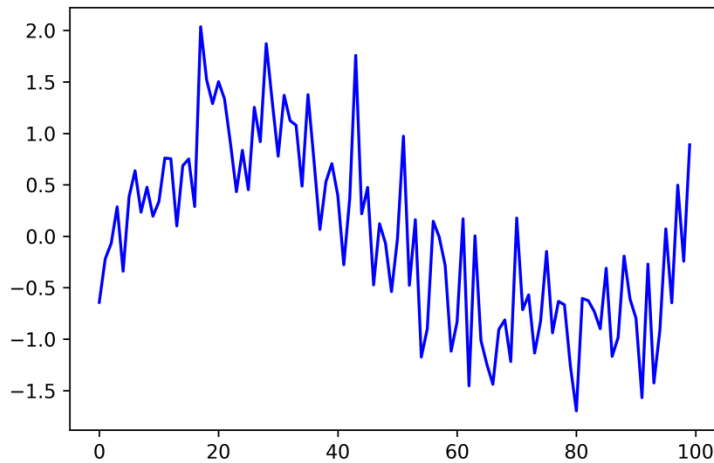
is a signal that changes when experiment is repeated (the world is noisy).

Example: $\sin(t) + 0.5 * \text{noise}$

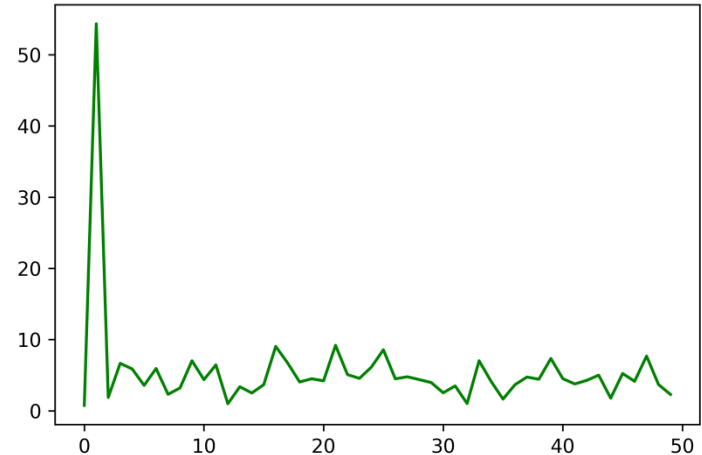
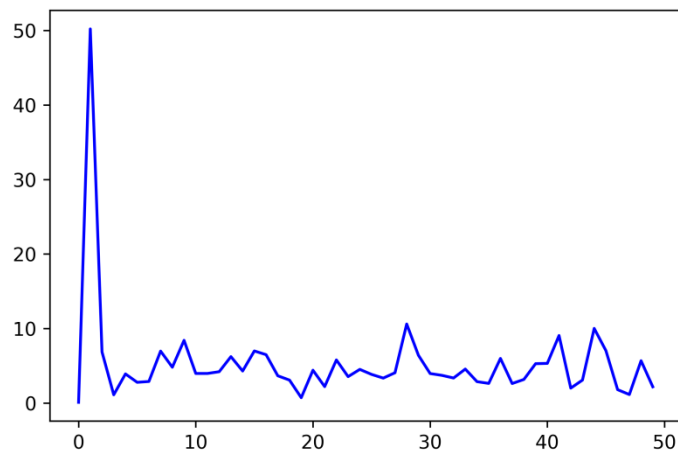


Random signal

$$\sin(t) + 0.5 * \text{noise}$$

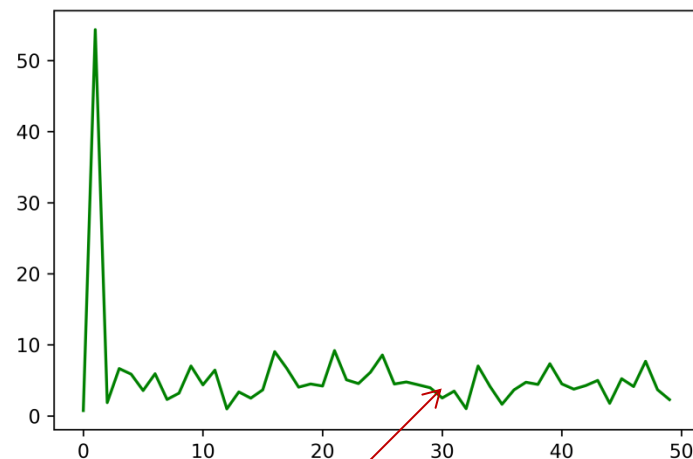
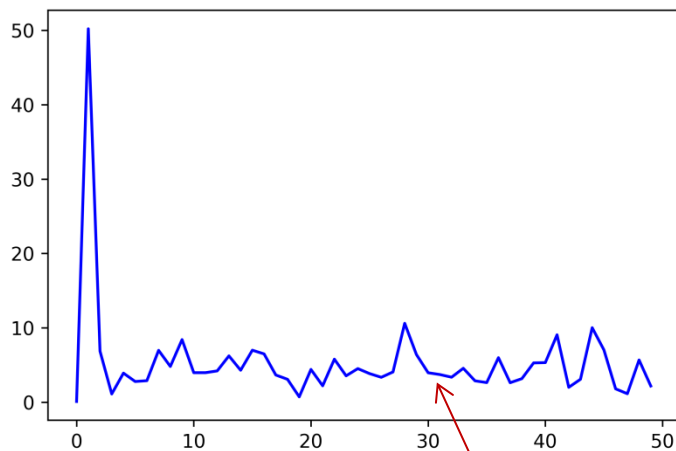


Fourier Transforms



Random signal

Fourier Transforms

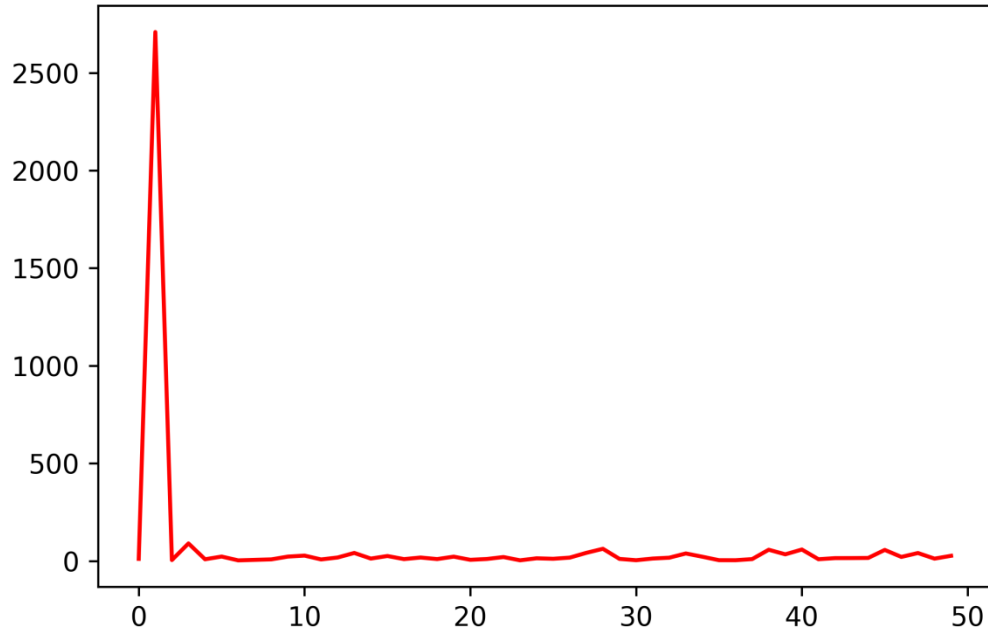


How often do I get a spike in that frequency?
What are the chances? How tall is it usually?
How much energy/power is in that frequency?

Spectral density

Energy spectral density (if we had K experiments):

$$S_E(f) = \frac{1}{K} \sum_{i=1}^K |X_i(f)|^2 = \frac{1}{K} \sum_{i=1}^K |\mathcal{F}(x_i(t))|^2$$



Simulation

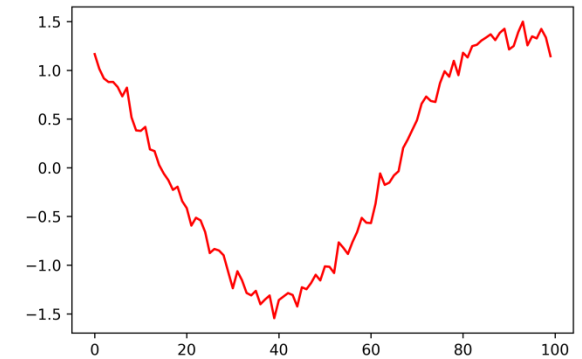
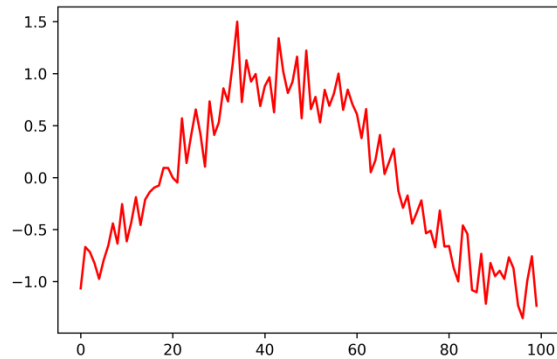
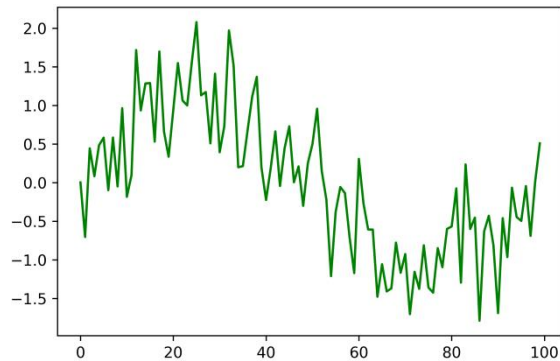
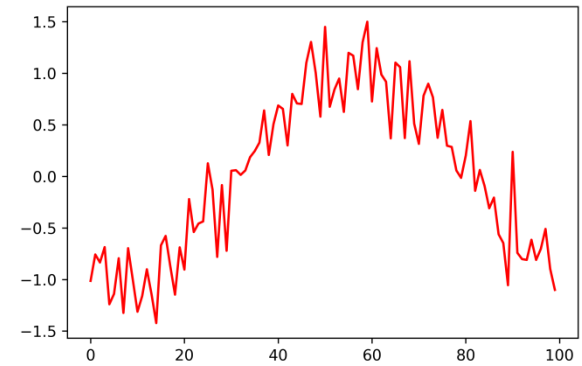
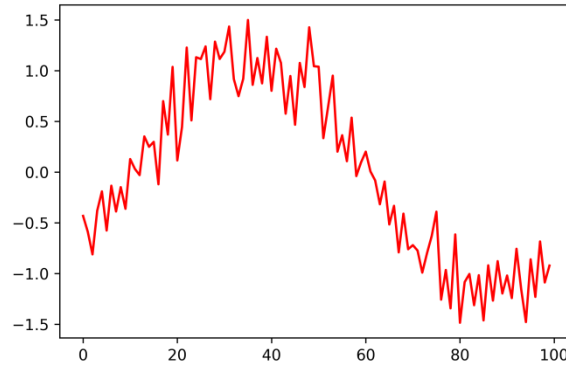
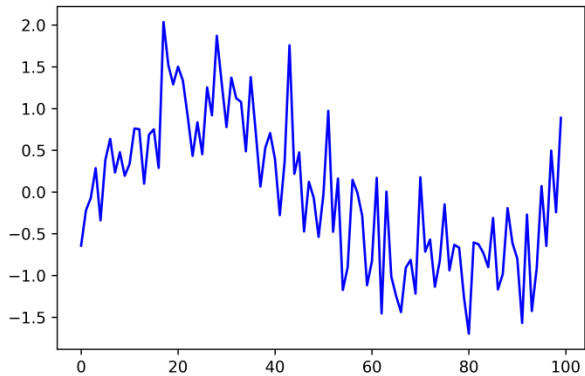
Spectral density (energy or power) can be used to simulate random signals. One way is by using shaping filters:

- Generate white noise
- Get its Fourier transform
- Multiply this transform with the ESD (filter the spectrum of the noise through the ESD)
- Get an inverse transform of the result

Shaping filters simulation

Real

Simulated from ESD



Spectral simulation

We generate Fourier coefficients:

$$X_F(k) = \sqrt{\frac{S_E\left(\frac{2\pi}{T}k_1\right)}{T^2(1 + \lambda_k^2)}} = \frac{1}{T} \sqrt{\frac{S_E\left(\frac{2\pi}{T}k_1\right)}{1 + \lambda_k^2}},$$

$k = 0, 1, \dots, M \geq N.$

Then those coefficients provide the simulated signal:

$$x(i) = \sum_{k=0}^{\frac{N}{2}} X_F(k) \exp\left[j2\pi\left(\frac{ki}{N}\right)\right], i \in [0, N).$$

This is simulation in Fourier basis, you can use others: Hartley, etc.

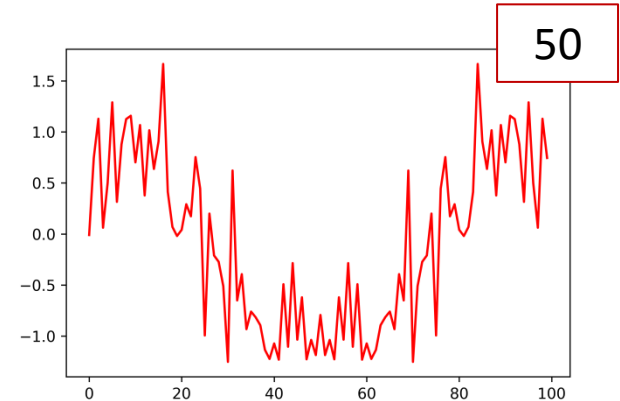
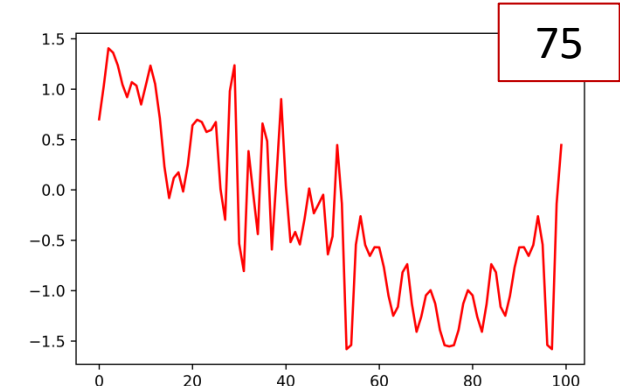
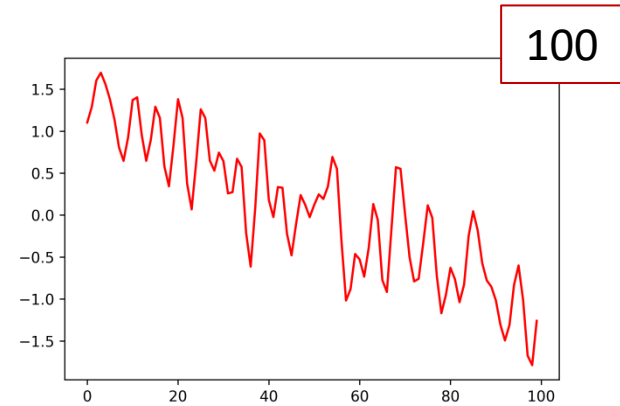
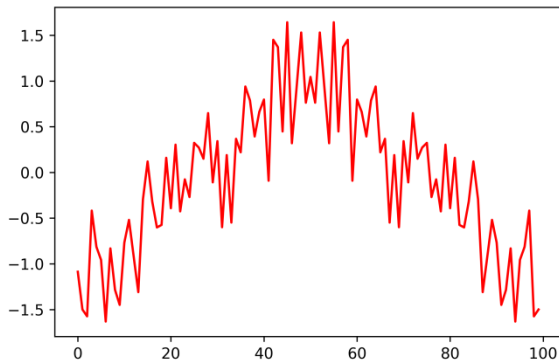
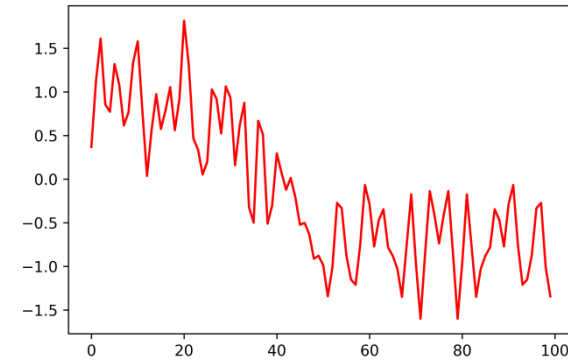
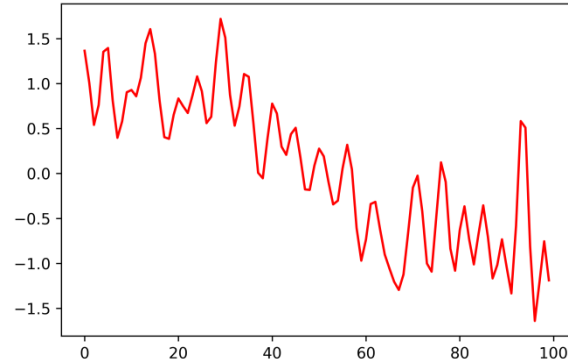
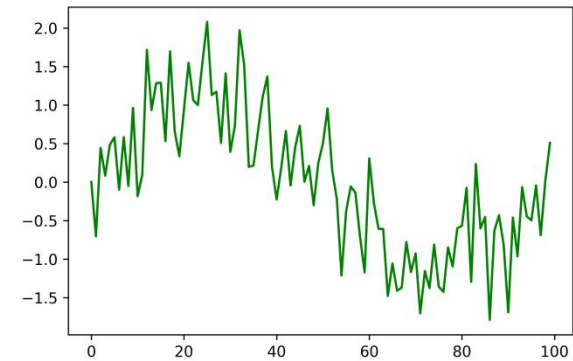
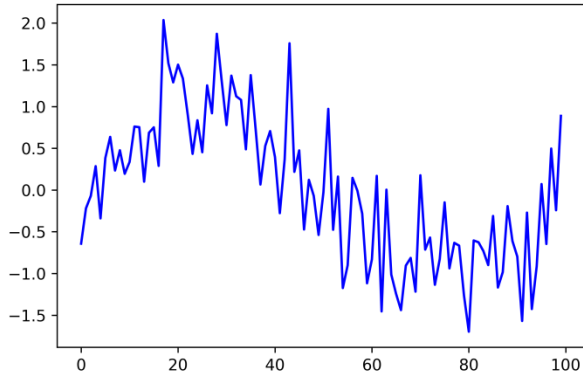
I might have cheated a bit...

```
XF = np.sqrt(ESD)
XF /= XF.max() # just normalized sqrt(ESD)
# turning noise into random signs
n3 = np.random.normal(0,1,50)
n3 /= abs(n3)
import cmath
z = np.empty([100], dtype=complex)
for i in range(0, 100):
    z[i] = 0
    for k in range(0, 50):
        # adding random signs to get random signals
        z[i] += XF[k] * n3[k] * cmath.exp(1j * k * i * math.pi/100)
```

Spectral simulation

Real

Simulated from ESD



2D signals

What kinds of data?

- 2D table data
- Spatial data
- Images
- Graphs

What changes?

- 1D space + time or 2D space (t_1, t_2 instead of t)
- Spatial frequencies

Computers “see” with help of image processing. Graphs may represent semantic webs. DSP gets close to AI.

What changes:

Fourier transform:

$$\mathcal{F}(x(t_1, t_2)) = X(f_1, f_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x(t_1, t_2) e^{-2\pi j(f_1 t_1 + f_2 t_2)} dt_1 dt_2 .$$

Energy spectral density:

$$S_E(f_1, f_2) = \frac{1}{K} \sum_{i=1}^K |X_i(f_1, f_2)|^2 = \frac{1}{K} \sum_{i=1}^K |\mathcal{F}(x_i(t_1, t_2))|^2 .$$

What happens then:

We generate Fourier coefficients:

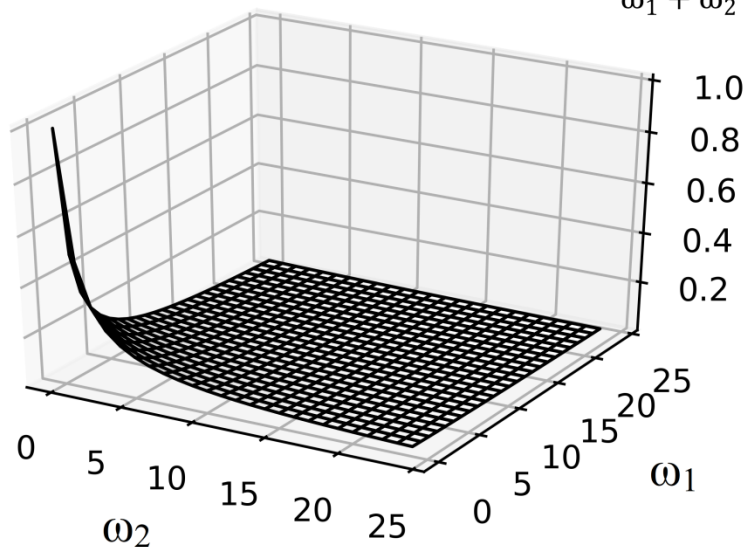
$$X_F(k_1, k_2) = \sqrt{\frac{S_E\left(\frac{2\pi}{T_1}k_1, \frac{2\pi}{T_1}k_2\right)}{T_1^2 T_2^2 (1 + \lambda_{k_1, k_2}^2)}} = \frac{1}{T_1 T_2} \sqrt{\frac{S_E\left(\frac{2\pi}{T_1}k_1, \frac{2\pi}{T_1}k_2\right)}{1 + \lambda_{k_1, k_2}^2}},$$
$$k_1, k_2 = 0, 1, \dots,$$

Then those coefficients provide the simulated signal:

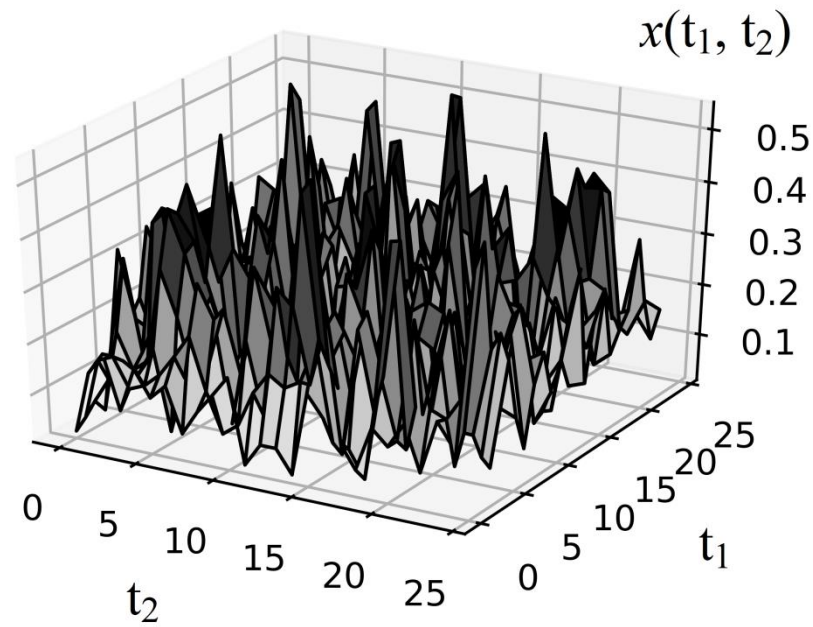
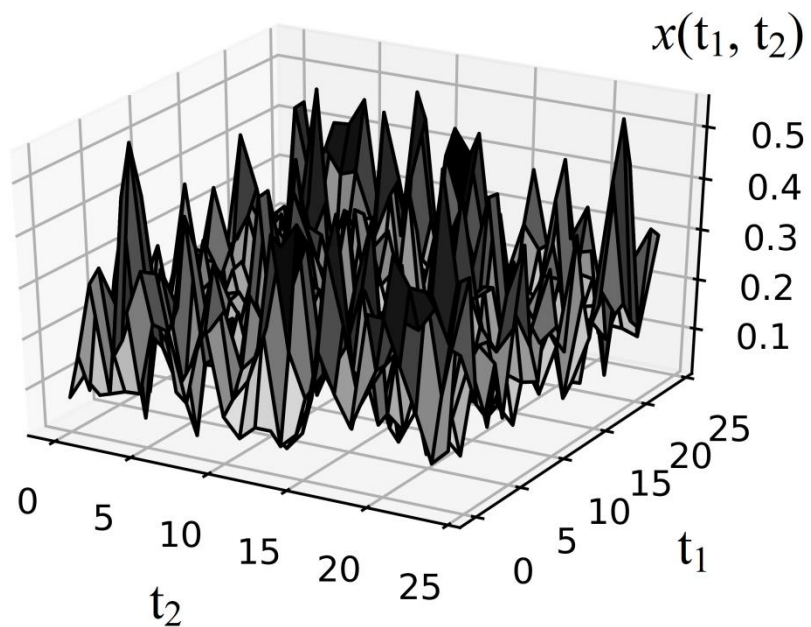
$$x(i_1, i_2) = \sum_{k_1=0}^{\frac{N_1}{2}} \sum_{k_2=0}^{\frac{N_2}{2}} X_F(k_1, k_2) \exp\left[j2\pi\left(\frac{k_1 i_1}{N_1} + \frac{k_2 i_2}{N_2}\right)\right], i_1 \in [0, N_1), i_2 \in [0, N_2).$$

ESD

$$S_E(\omega_1, \omega_2) = \frac{1}{\omega_1 + \omega_2 + 1}$$

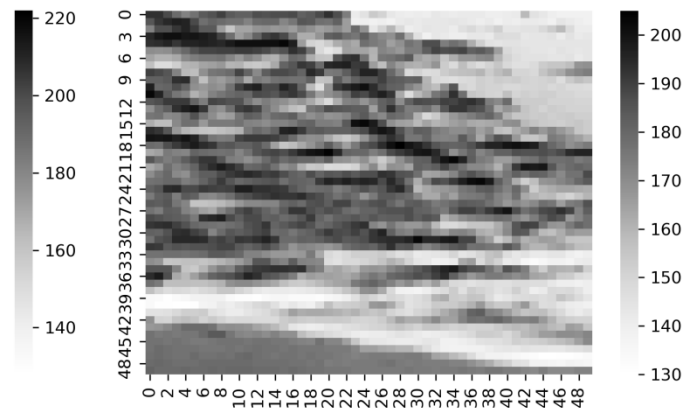
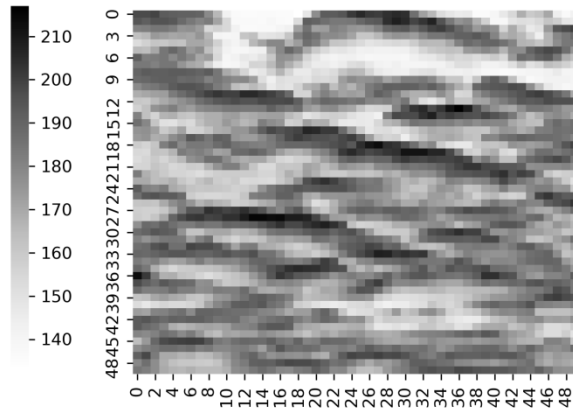
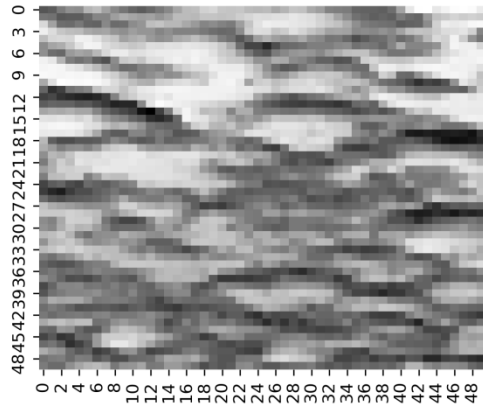


Random Signals in Fourier basis

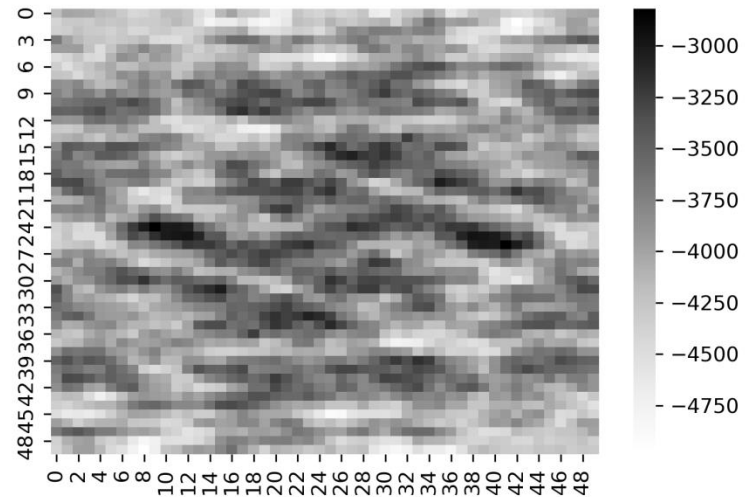
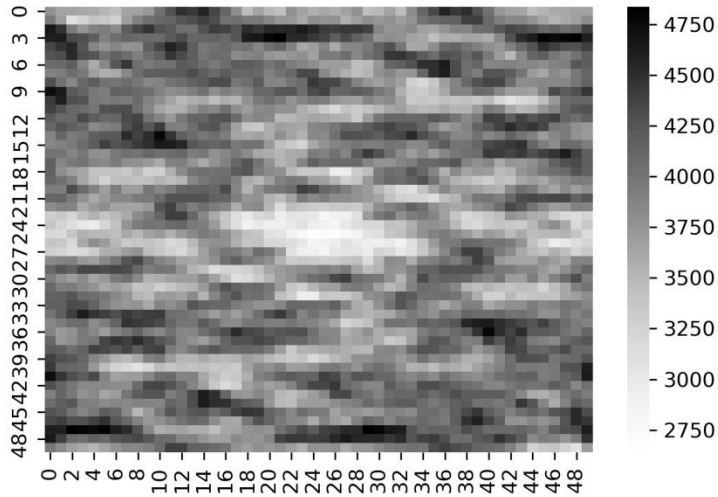


Clouds

Real

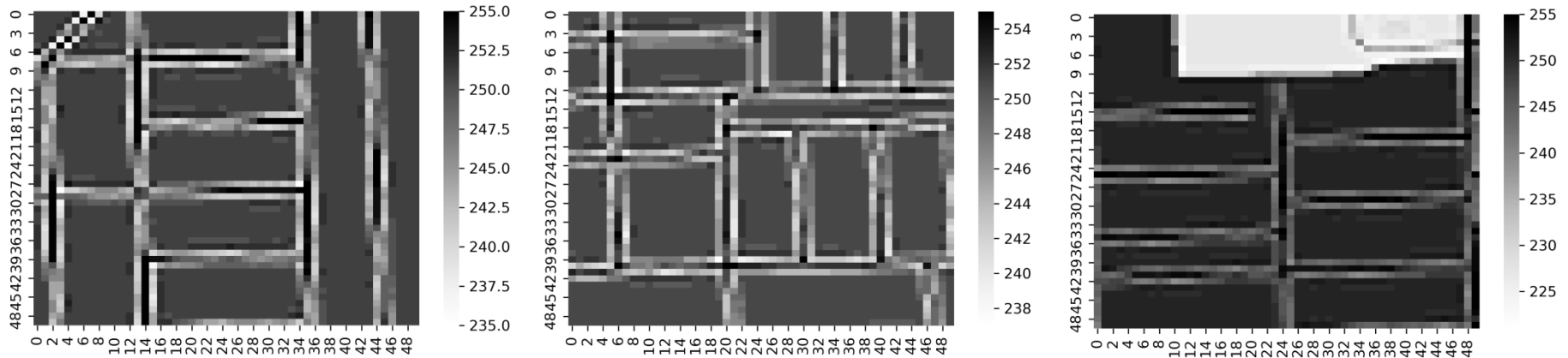


Fourier simulated

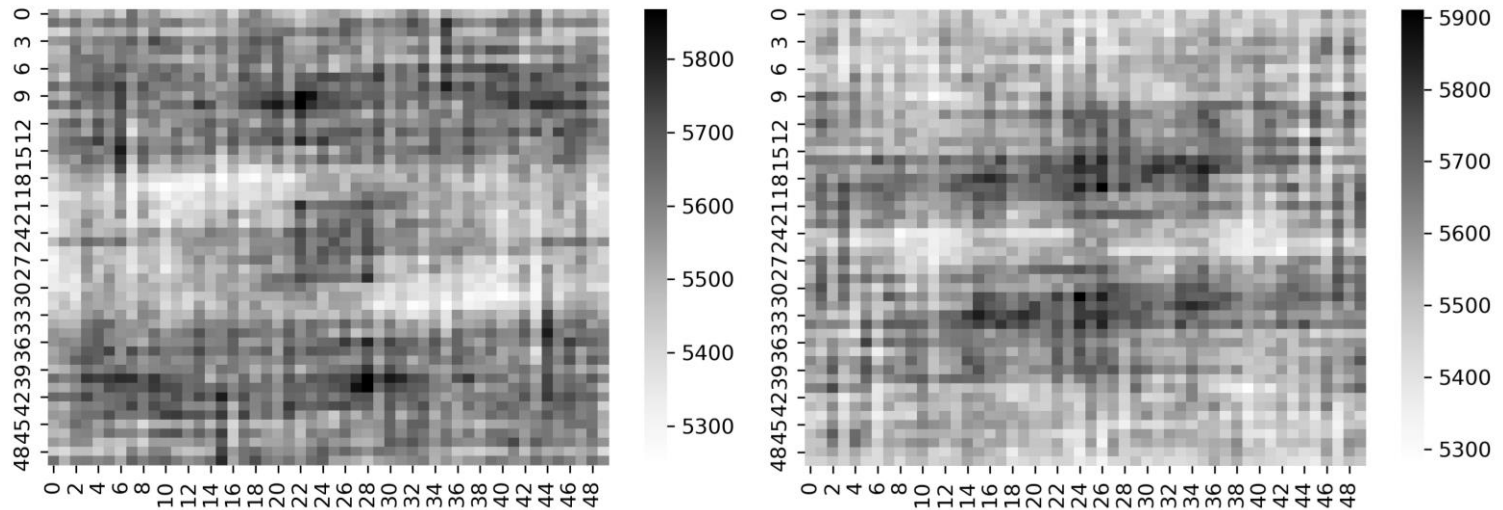


Streets

Real



Fourier simulated



A decorative vertical border on the left side of the slide, featuring a dark grey background with clusters of bright orange roses and small white flowers with yellow centers.

Thanks!