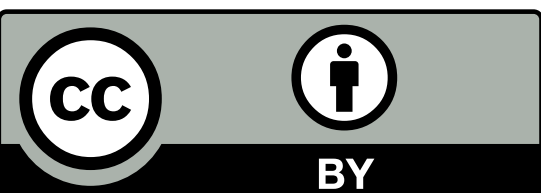


Spectral Analysis

Digital Audio Coding



SEBASTIEN.BOISGERAULT@MINES-PARISTECH.FR

Discrete-Time Fourier Transform

$$x(t) = \int_{-\Delta f/2}^{\Delta f/2} x(f) \exp(i2\pi f t) df$$

$$x(t) \xrightleftharpoons[\mathcal{F}^{-1}]{\mathcal{F}} x(f)$$

time domain

$$t \in \mathbb{Z}\Delta t$$

$$\Delta t = 1/\Delta f$$

frequency domain

$$f \in \mathbb{R}/\mathbb{Z}\Delta f$$

$$\Delta f = 1/\Delta t$$

$$x(f) = \Delta t \sum_{t \in \mathbb{Z}\Delta t} x(t) \exp(-i2\pi f t)$$

Spectral Decomposition Problem

Given $x(t) : \mathbb{Z}\Delta t \rightarrow \mathbb{R}$,

find $\left| \begin{array}{l} a(f) : \mathbb{R} \rightarrow \mathbb{R}_+ \\ \phi(f) : \mathbb{R} \rightarrow [-\pi, \pi) \end{array} \right.$

such that $\forall t \in \mathbb{Z}\Delta t,$

$$x(t) = \int_0^{+\infty} a(f) \cos(2\pi f t + \phi(f)) df$$

Complex Exponentials

Instead, search for $x(f) : \mathbb{R} \rightarrow \underline{\mathbb{C}}$ such that

$$x(t) = \int_{\underline{-\infty}}^{+\infty} x(f) \exp(i2\pi ft) df$$

(with $x(-f) = \overline{x(f)}$)

$$\text{then} \left| \begin{array}{l} a(f) = 2|x(f)| \\ \phi(f) = \angle x(f) \end{array} \right.$$

Uniqueness of the Decomposition

Define $\Delta f = \frac{1}{\Delta t}$

If $x(f) : \mathbb{R} \rightarrow \mathbb{C}$ **is a solution of**

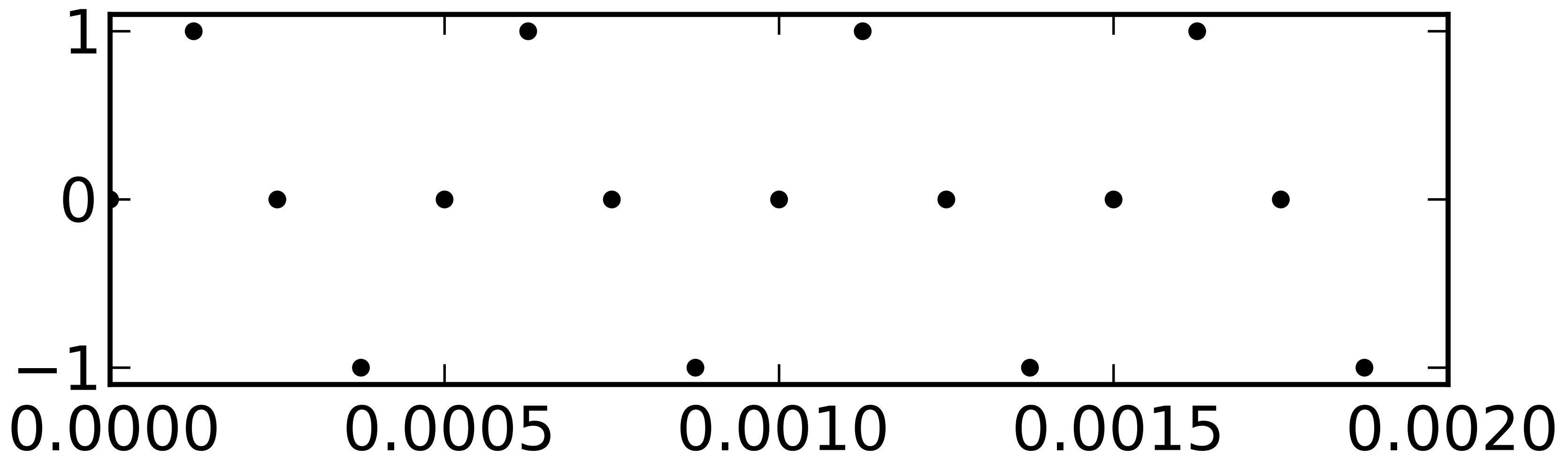
$$x(t) = \int_{-\infty}^{+\infty} x(f) \exp(i2\pi ft) df$$

... so is $x(f - k\Delta f)$ **for any** $k \in \mathbb{Z}$.

?

Frequency Ambiguity

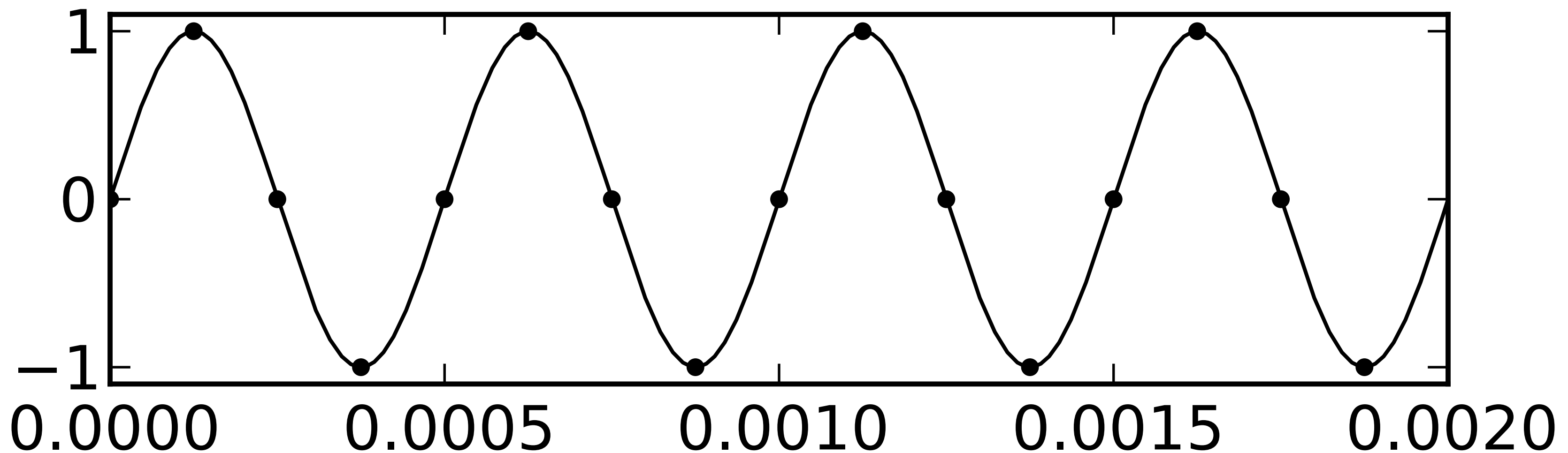
What is the frequency of this signal ?



$$\Delta f = 8000 \text{ Hz}$$

Frequency Ambiguity

What is the frequency of this signal ?

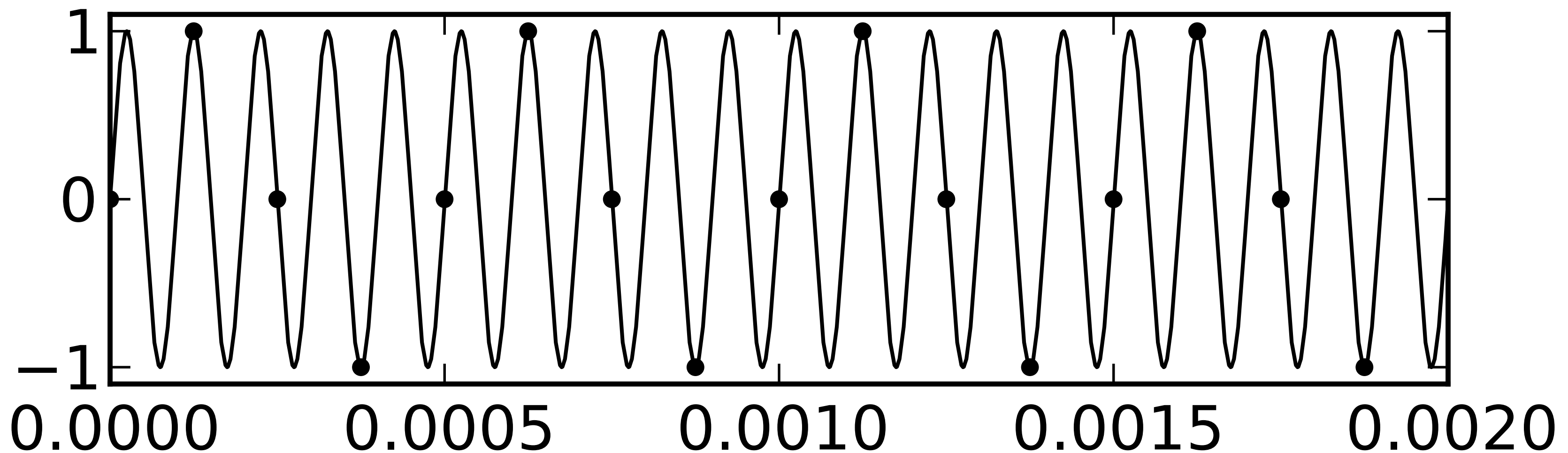


$$f = 2000 \text{ Hz}$$

$$\Delta f = 8000 \text{ Hz}$$

Frequency Ambiguity

What is the frequency of this signal ?



$$f = 10000 \text{ Hz}$$

$$\Delta f = 8000 \text{ Hz}$$

Spectral Decomposition - Conclusion

Search for $x(f) : \mathbb{R} \rightarrow \mathbb{C}$, solution of

$$x(t) = \int_{-\Delta f/2}^{+\Delta f/2} x(f) \exp(i2\pi f t) df$$

and enforce Δf –periodicity of $x(f)$.

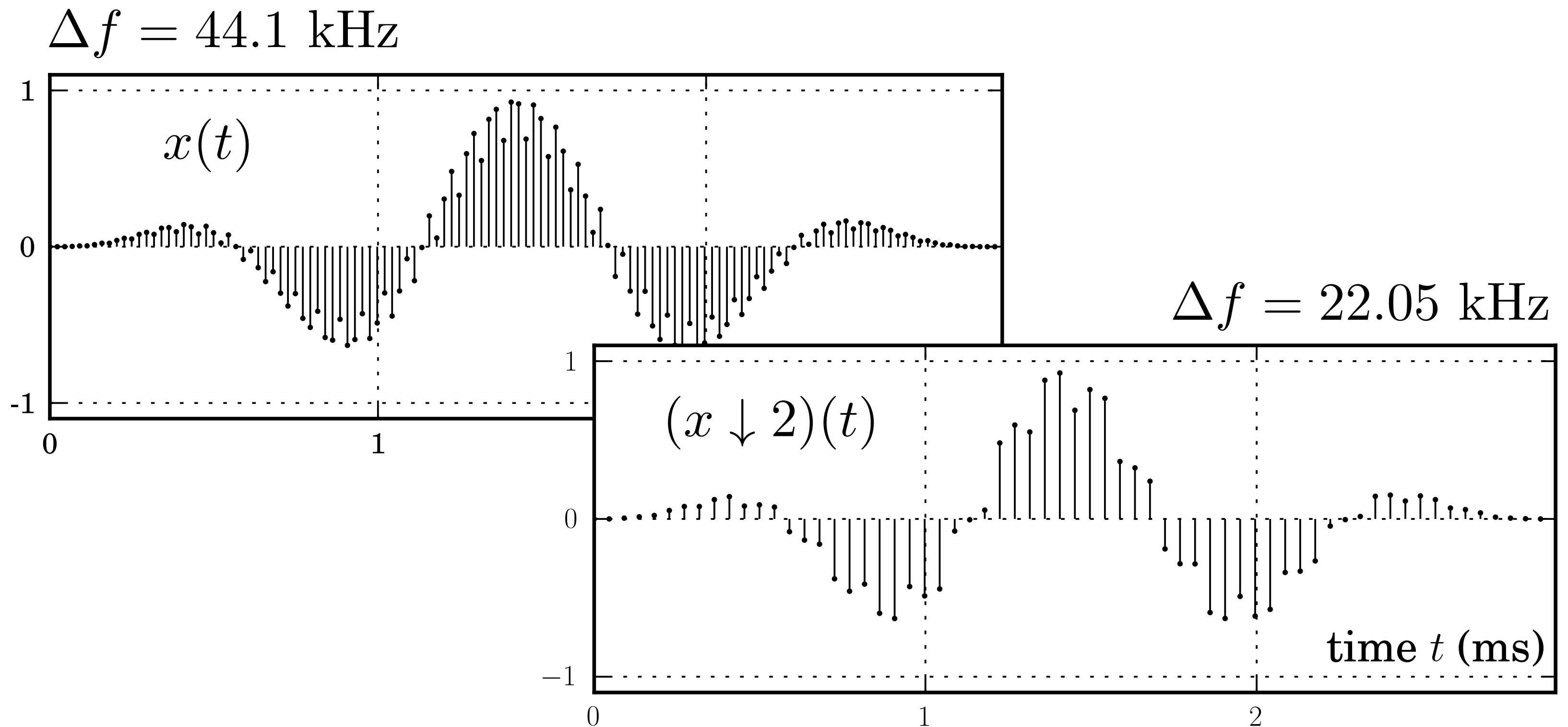
These choices yield uniqueness and:

$$x(f) = \Delta t \sum_{t \in \mathbb{Z}\Delta t} x(t) \exp(-i2\pi f t)$$

Decimation

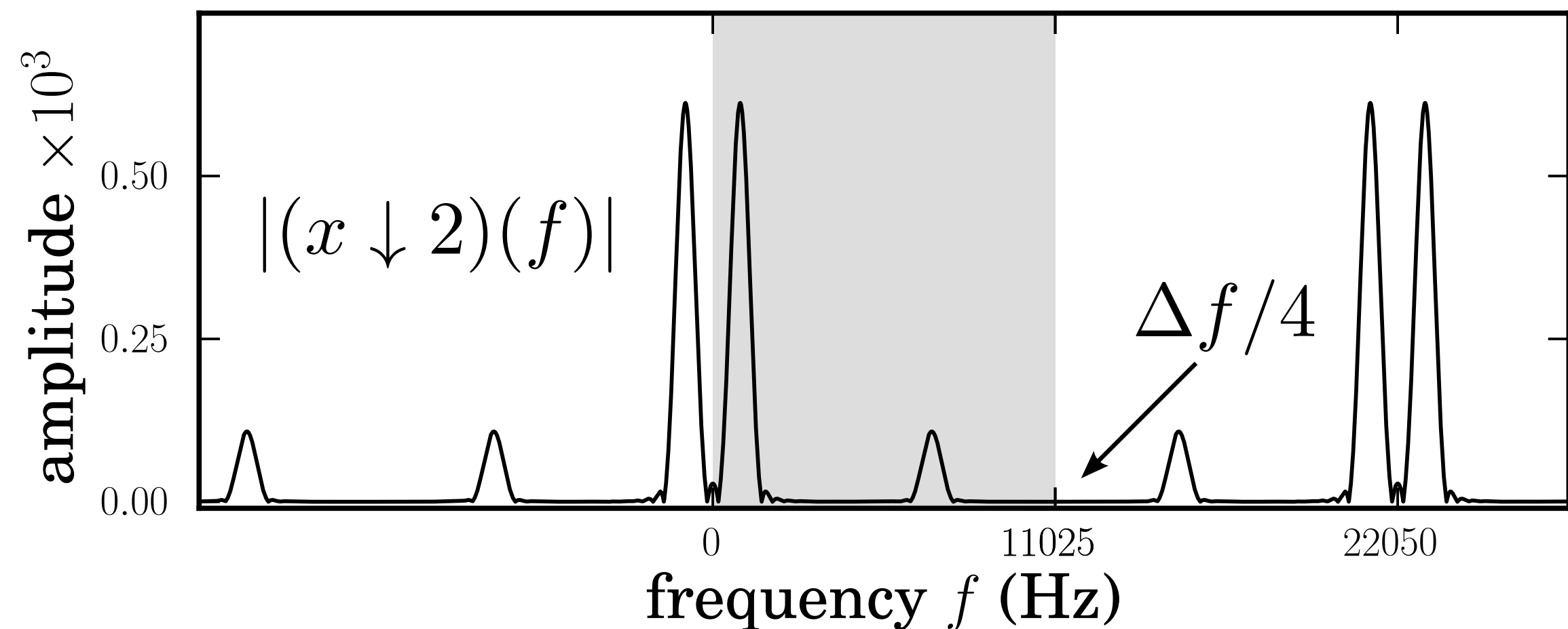
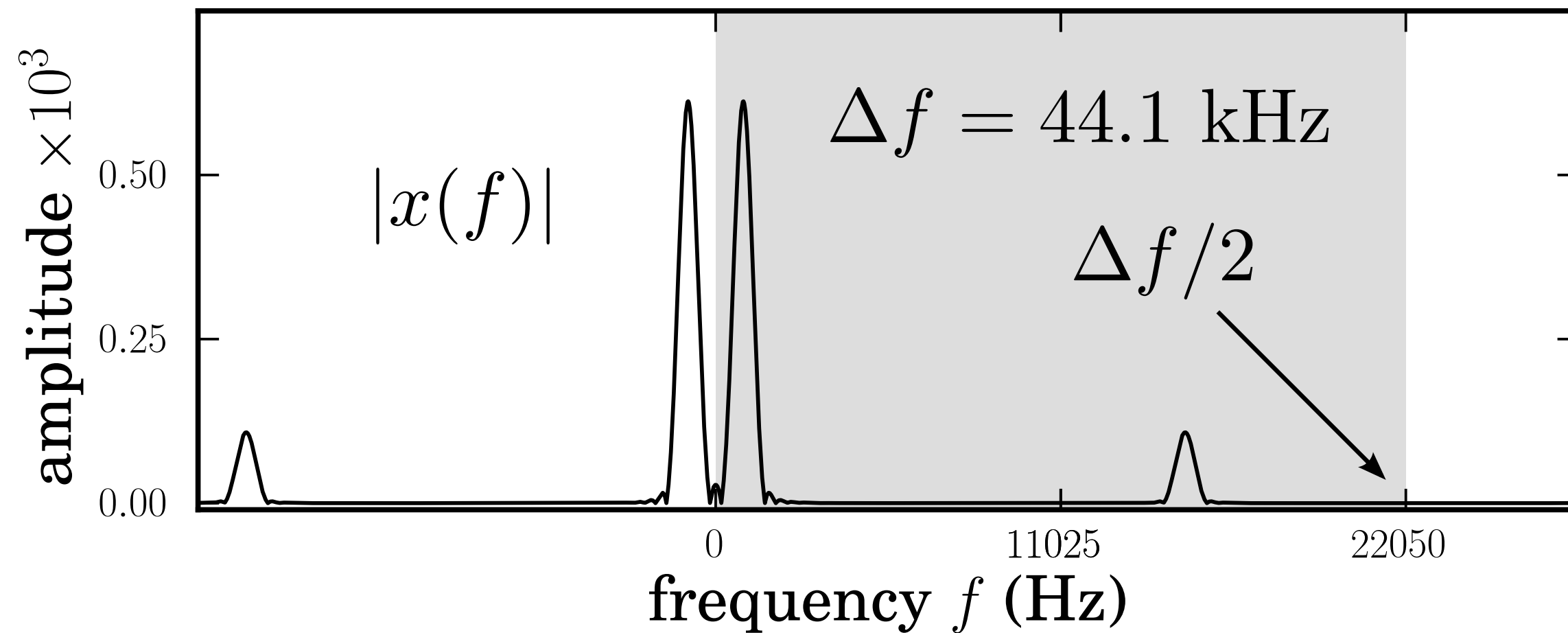
$$x : \mathbb{Z}\Delta t \rightarrow \mathbb{R} \longrightarrow x \downarrow M : \mathbb{Z}M\Delta t \rightarrow \mathbb{R}$$

$$(x \downarrow M)(t) = x(t), \quad t \in \mathbb{Z}M\Delta t$$



Decimation - Spectrum

$$(x \downarrow 2)(f) = x(f) + x(f + \Delta f / 2)$$



Filtering

It's multiplication in the frequency domain:

$$y(f) = h(f)u(f)$$

or signal **convolution** in the time domain:

$$y(t) = (h * u)(t) = \Delta t \sum_{t' \in \mathbb{Z}\Delta t} h(t')u(t - t')$$



$h(t)$ is the filter **impulse response**:

$$y(t) = h(t) \quad \text{if } u(t) = \begin{cases} 1/\Delta t & \text{if } t = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Low-Pass Filter

Defined in the frequency domain by:

$$h(f) = \begin{cases} 1 & \text{if } |f| \leq f_c \\ 0 & \text{otherwise} \end{cases} \quad \text{for } |f| \leq \Delta f / 2$$

where f_c is the **cutoff frequency**.

In the time domain:

$$h(t) = 2f_c \operatorname{sinc} 2f_c t \quad \text{with} \quad \operatorname{sinc} x = \frac{\sin \pi x}{\pi x}$$

The impulse response is **acausal** and **infinite**.

Low-Pass Filter

Concrete implementations are **causal** and **finite**:

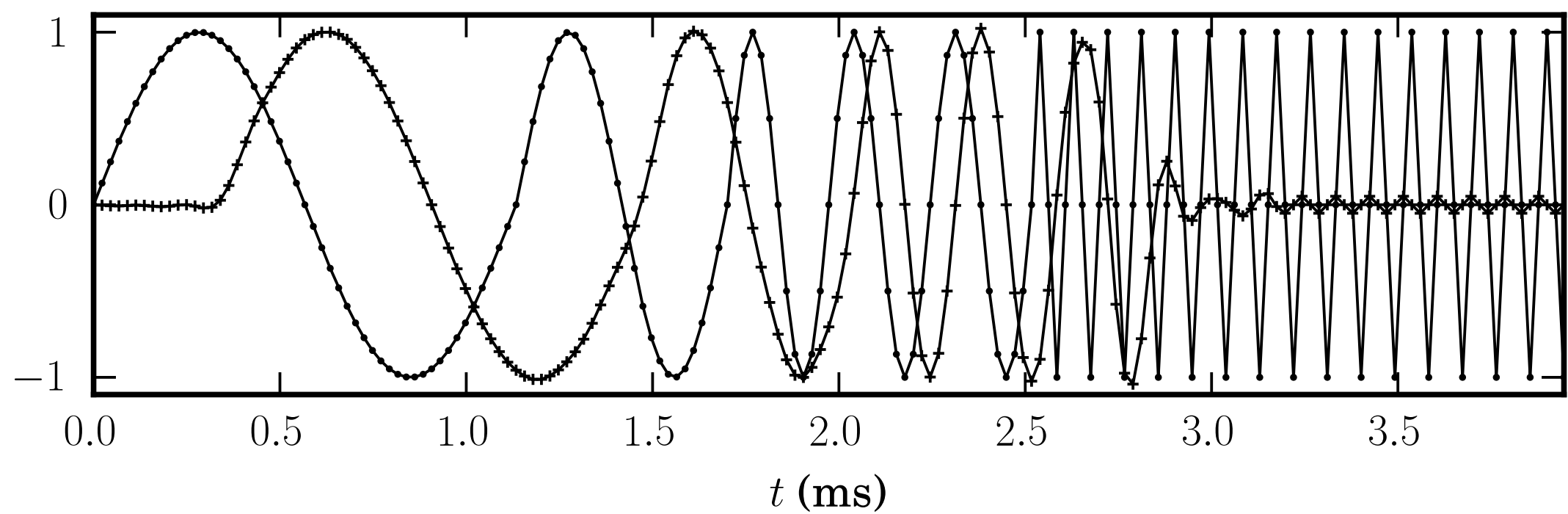
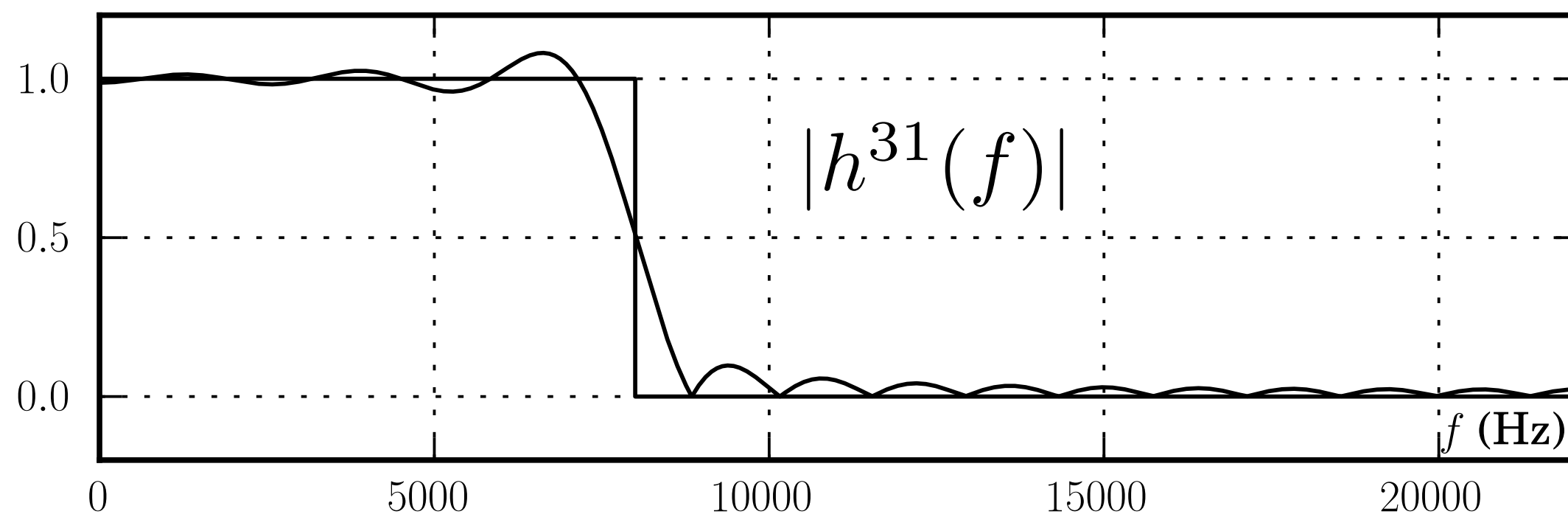
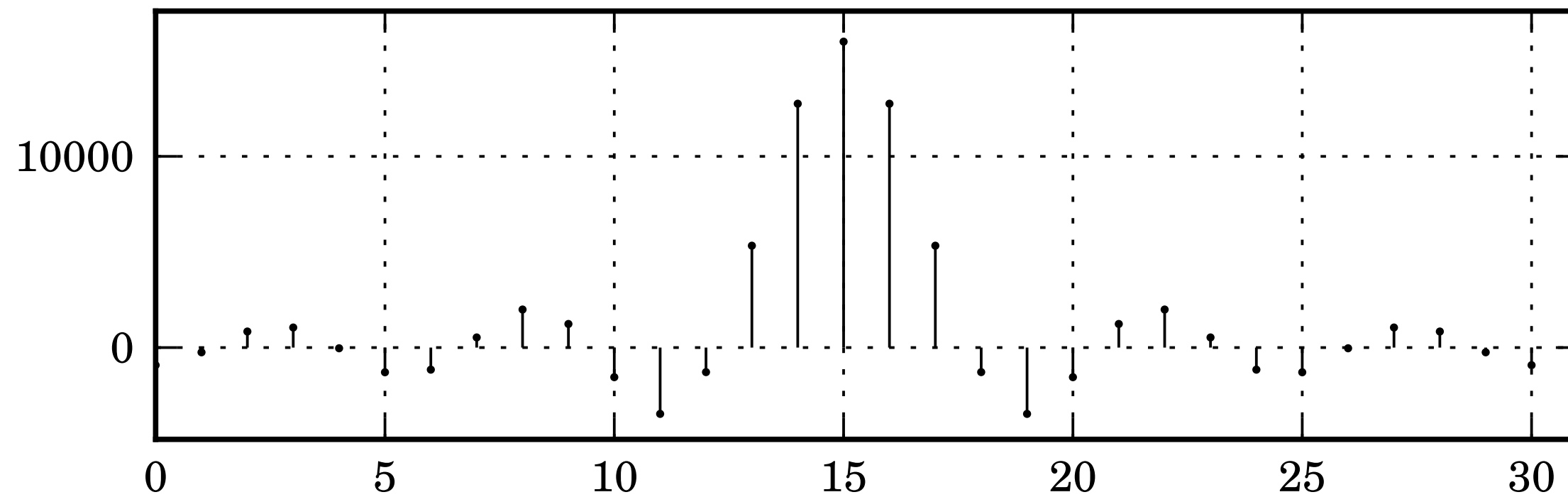
```
def low_pass(fc, dt=1.0, window=ones):  
    def h(n):  
        t = arange(-0.5 * (n-1), 0.5 * (n-1) + 1) * dt  
        return 2 * fc * sinc(2 * fc * t) * window(n)  
    return h
```

Filter the (finite, causal) signal $u(t)$:

```
>>> N = 31  
>>> h = low_pass(fc=8000.0, dt=1.0/44100.0)(N)  
>>> y = dt * convolve(h, u)
```

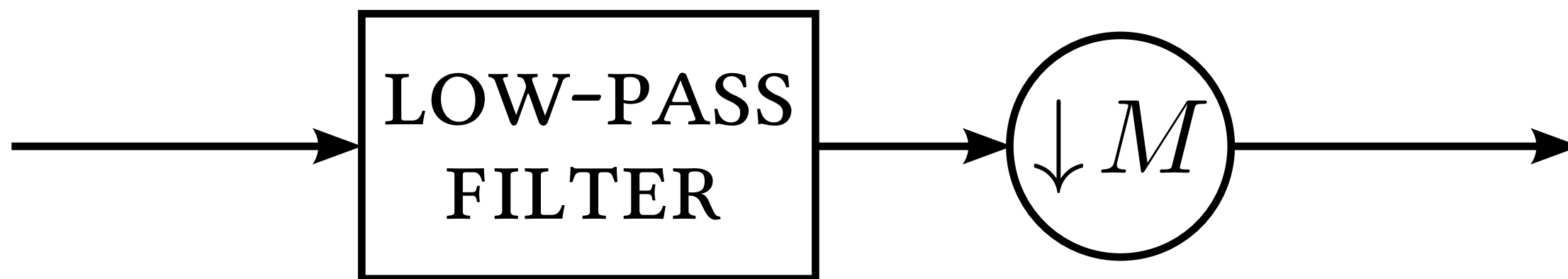
Low-Pass Filter

$$h_n^{31} = 16000 \operatorname{sinc} 16000(n - 15), \quad n \in \{0, \dots, 30\}$$



Downsampling

$$f_c = \frac{\Delta f}{2M}$$



F.T. & Signal Energy

Parseval theorem for Fourier series yields:

$$\Delta t \sum_{t \in \mathbb{Z} \Delta t} |x(t)|^2 = \int_{-\Delta f/2}^{\Delta f/2} |x(f)|^2 df$$

and proves that \mathcal{F} is an isomorphism:

$$L^2(\mathbb{Z} \Delta t) \longleftrightarrow L^2(\mathbb{R} / \mathbb{Z} \Delta f)$$

F.T. & Measures

Some signals are not of finite energy but may be represented in the Fourier domain as **measures***:

$$x(f) = x_1(f) + \sum_k a_k \delta(f - f_k)$$

Annotations:

- $x_1(f)$ is labeled $L^1(\mathbb{R}/\mathbb{Z}\Delta f)$ with a line pointing to it.
- $\delta(f - f_k)$ is labeled *periodic* with a line pointing to it.
- $\sum_k |a_k| < +\infty$ is written below the sum, with a line pointing to the a_k term.

They correspond in the time domain to:

$$x(t) = x_1(t) + \sum_k a_k \exp(i2\pi f_k t)$$

*: periodic, complex-valued, with no singular continuous part.

Spectrum Computation

Finite causal signal: all values are zero but :

$$x(0), x(\Delta t), \dots, x((N-1)\Delta t)$$

$$[\mathcal{F}x](f) = \Delta t \sum_{n=0}^{N-1} x(n\Delta t) \exp(-i2\pi f n \Delta t)$$

```
def F(x, dt=1.0):
```

```
    n = reshape(r_[0:len(x)], (-1, 1))
```

```
    def Fx(f):
```

```
        f = reshape(f, (1, -1))
```

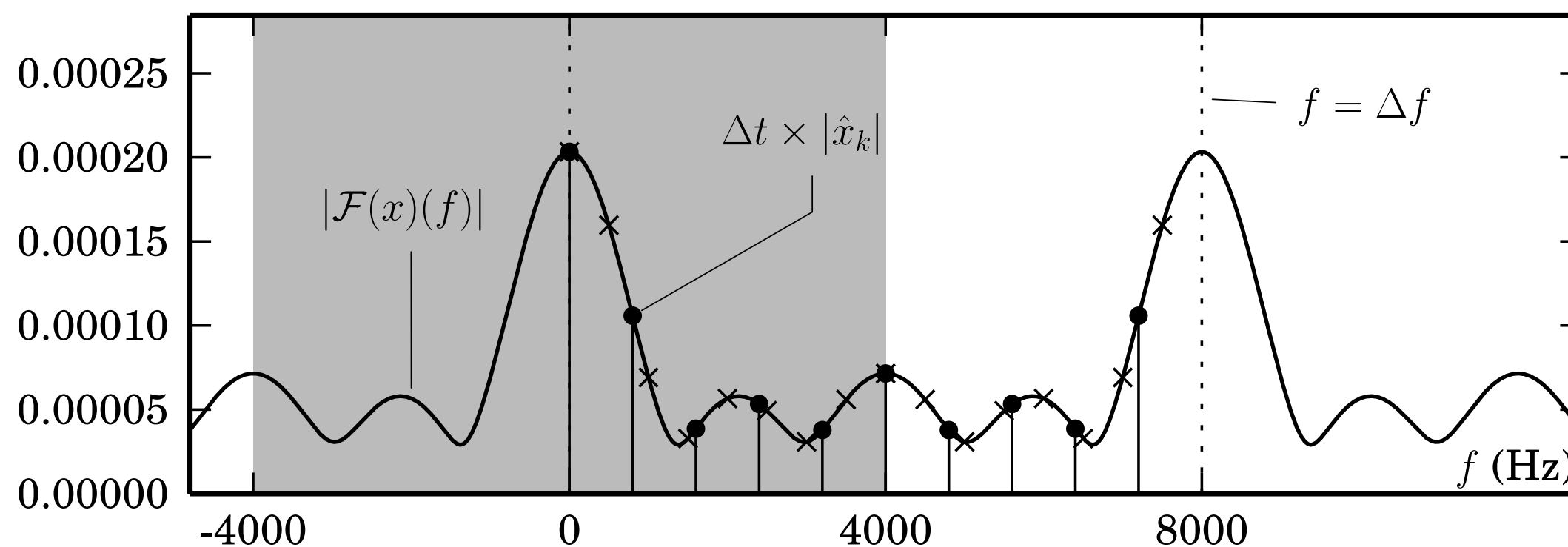
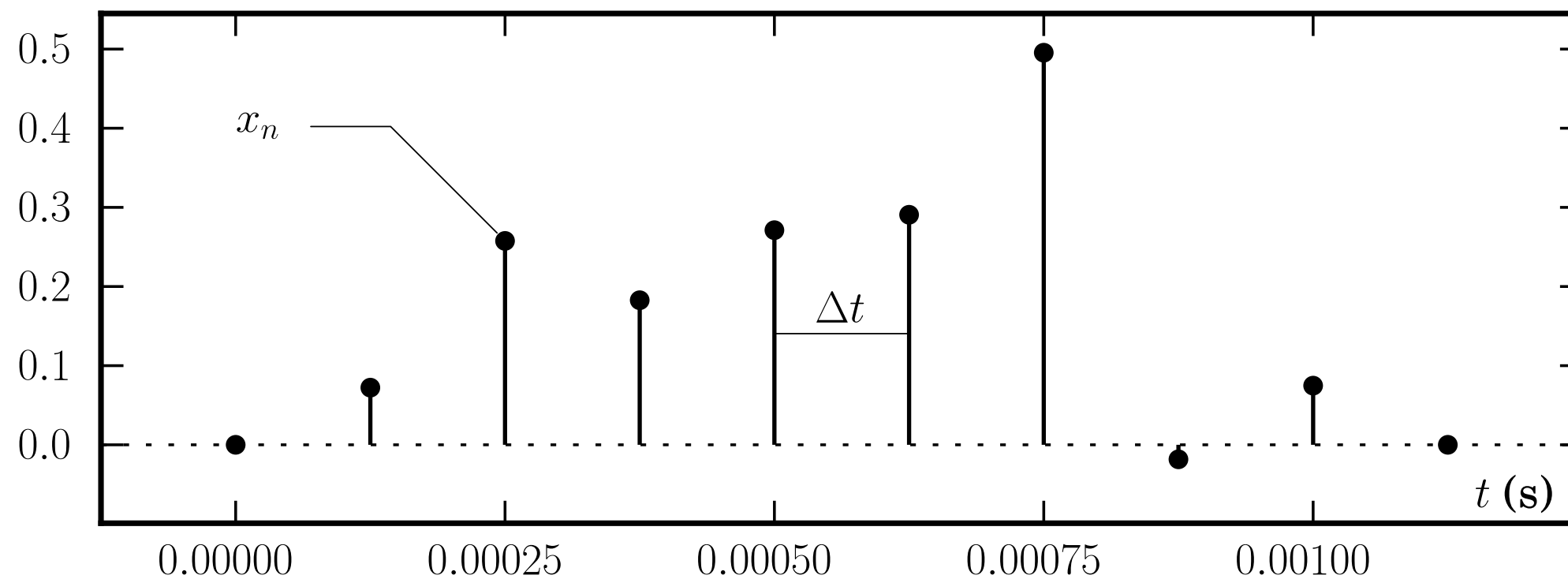
```
        return dt * dot(x, exp(-1j * 2 * pi * dt * n * f))
```

```
    return Fx
```

Discrete Fourier Transform - DFT

$$x_n = x(t = n\Delta t) \longrightarrow \hat{x}_k = \Delta f \times x(f = k\Delta f/N)$$

$$n = 0, 1, \dots, N-1 \qquad k = 0, 1, \dots, N-1$$



zero-padding increases frequency resolution

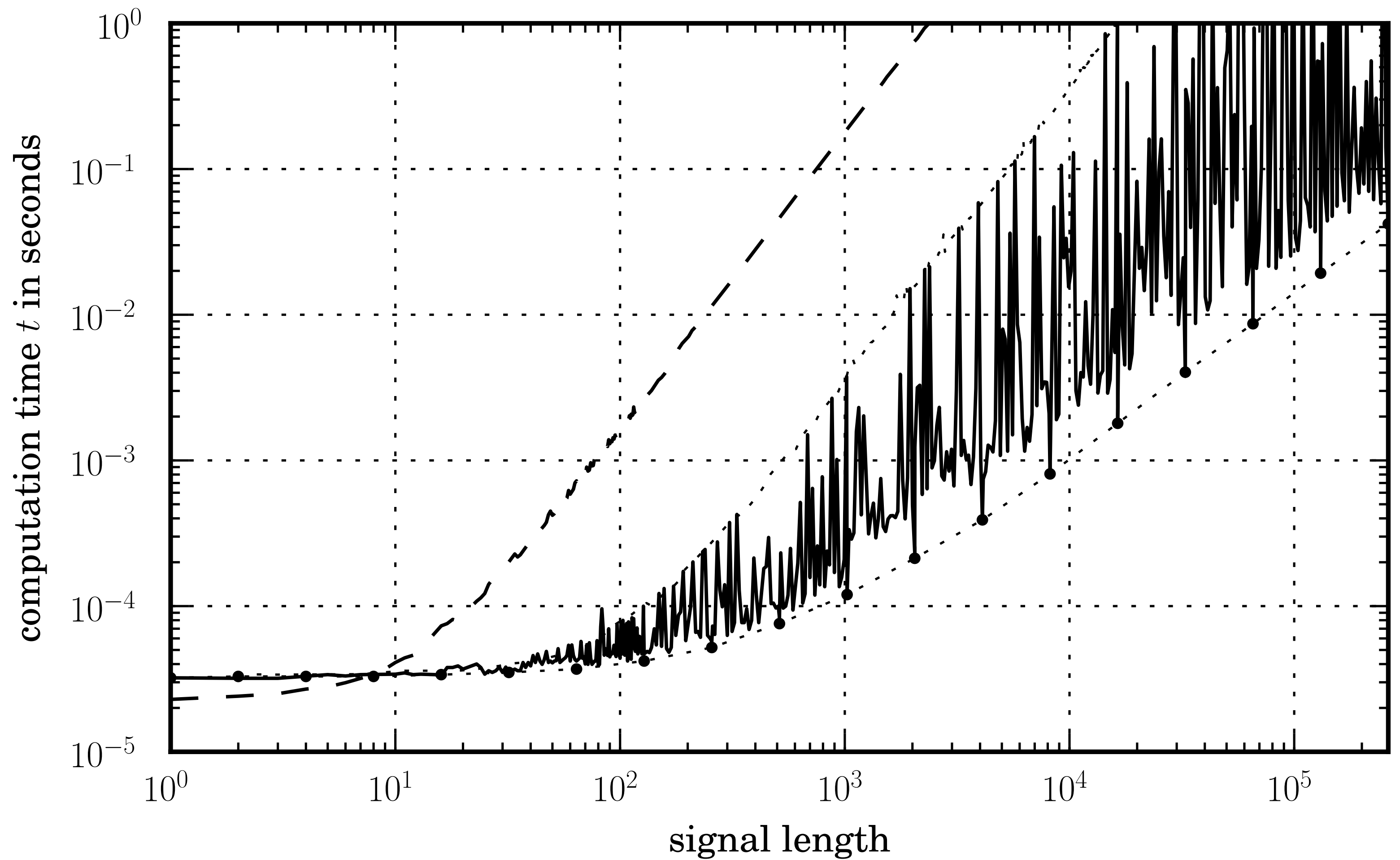
Fast Fourier Transform - FFT

The DFT is a **function**, not an algorithm, that may be used to compute some spectrum (a.k.a. (DT)FT) values, for finite and causal signals.

The naive algorithm (as a matrix-vector product) to compute the DFT has a $\mathcal{O}(N^2)$ complexity.

FFT refer to a **family of algorithms** that perform the DFT with a $\mathcal{O}(N \log N)$ complexity.

FFT Benchmarks

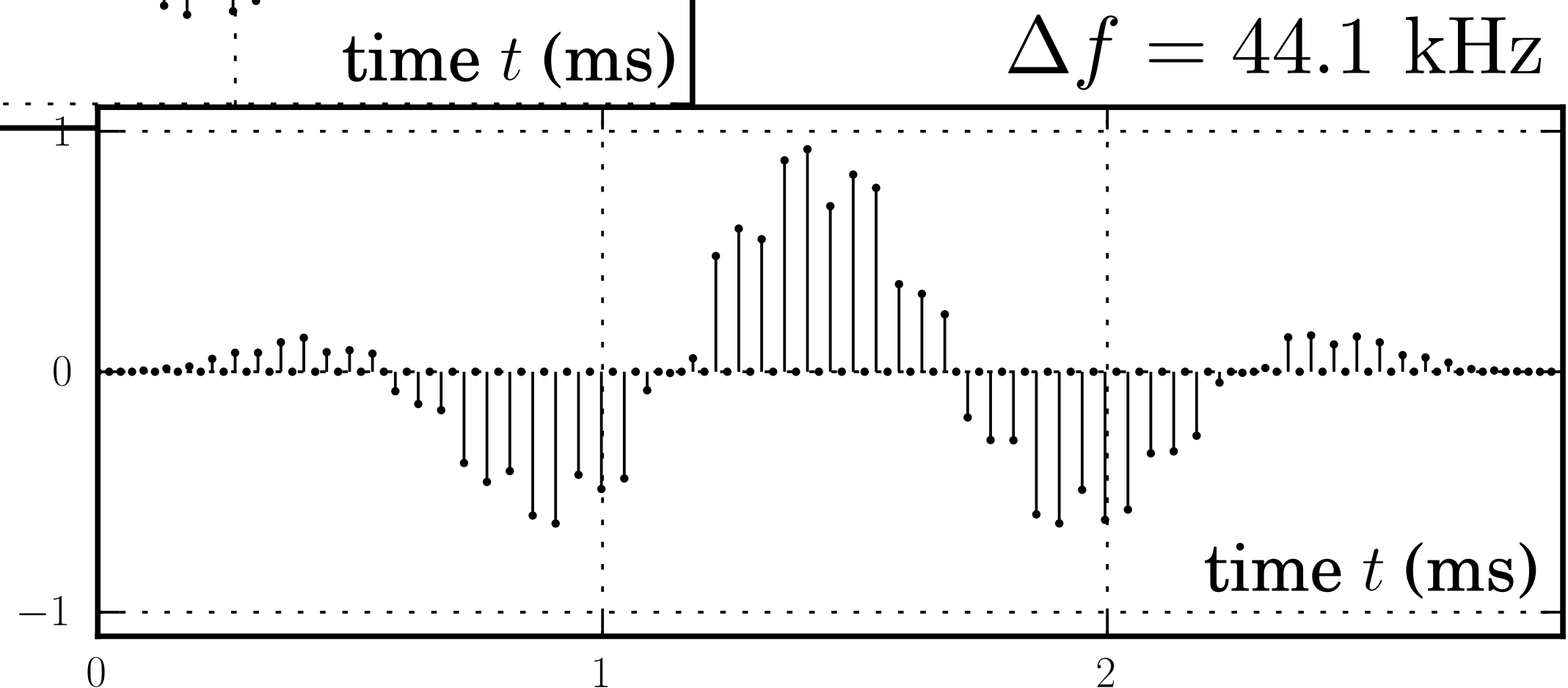
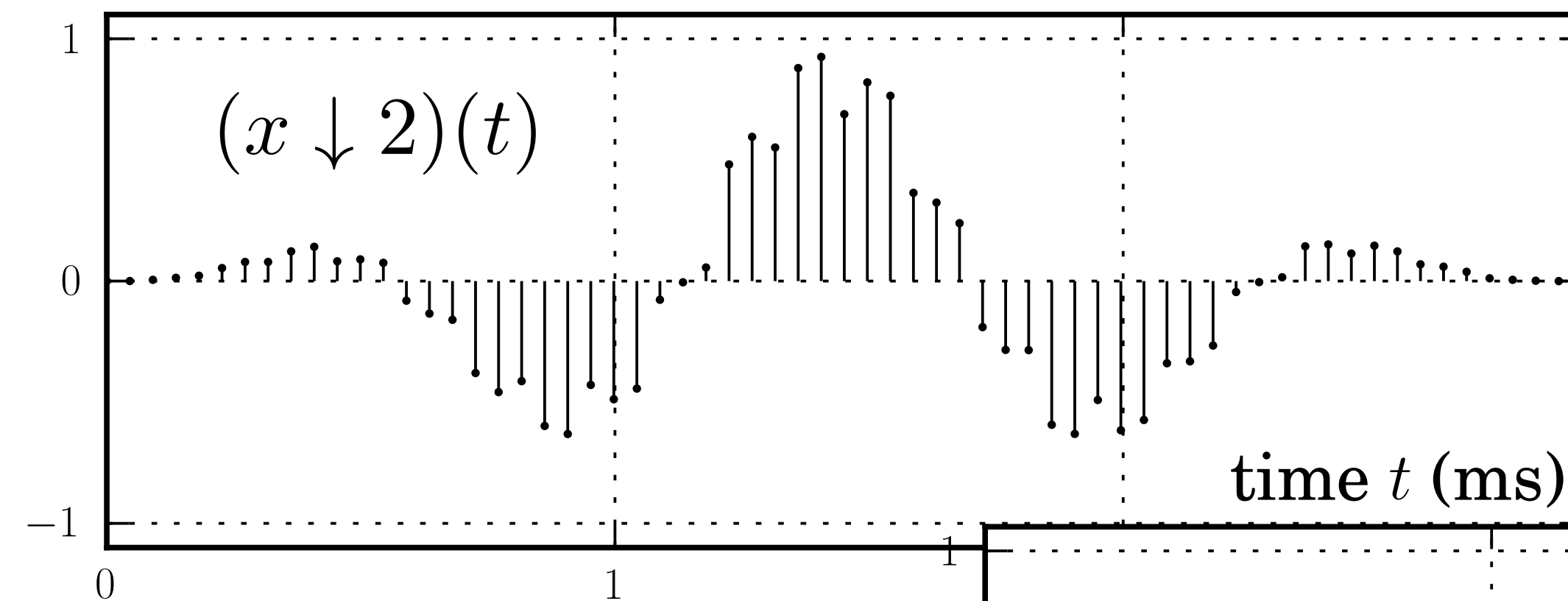


Expansion

$$x : \mathbb{Z}\Delta t \rightarrow \mathbb{R} \longrightarrow x \uparrow M : \mathbb{Z}\Delta t/M \rightarrow \mathbb{R}$$

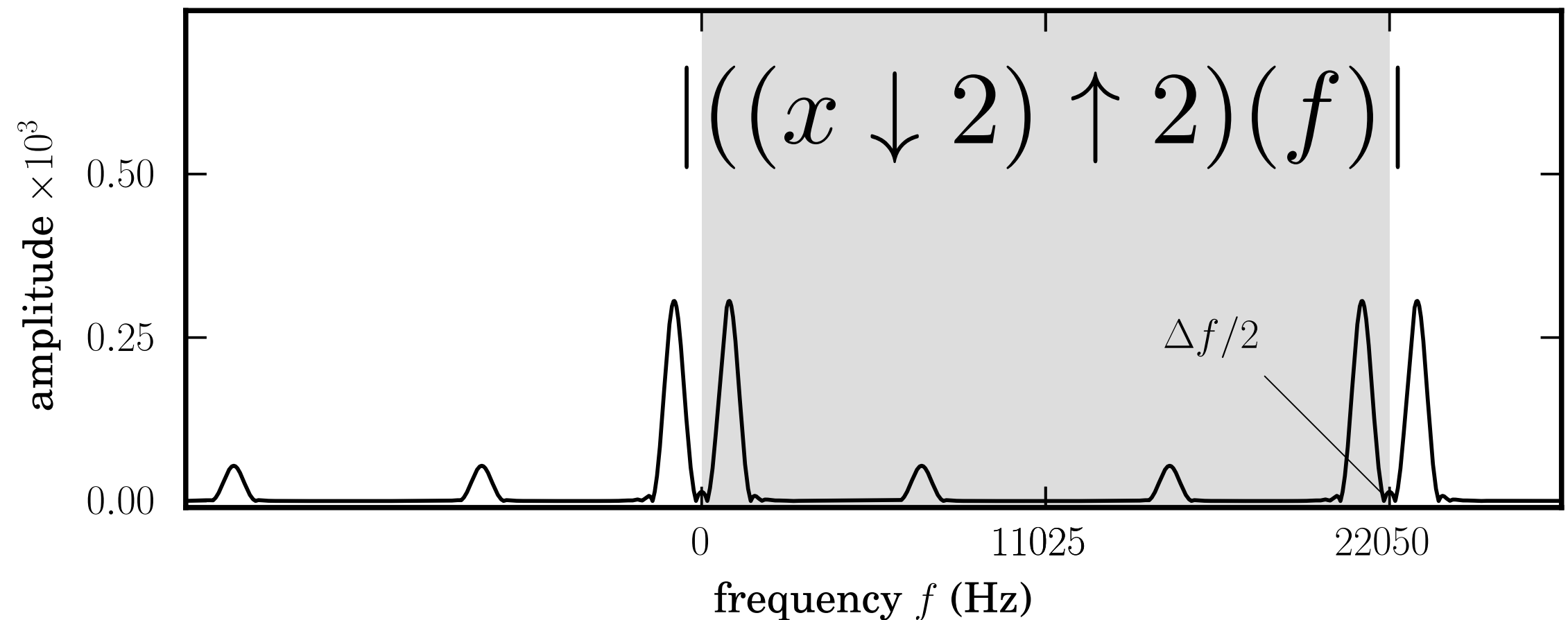
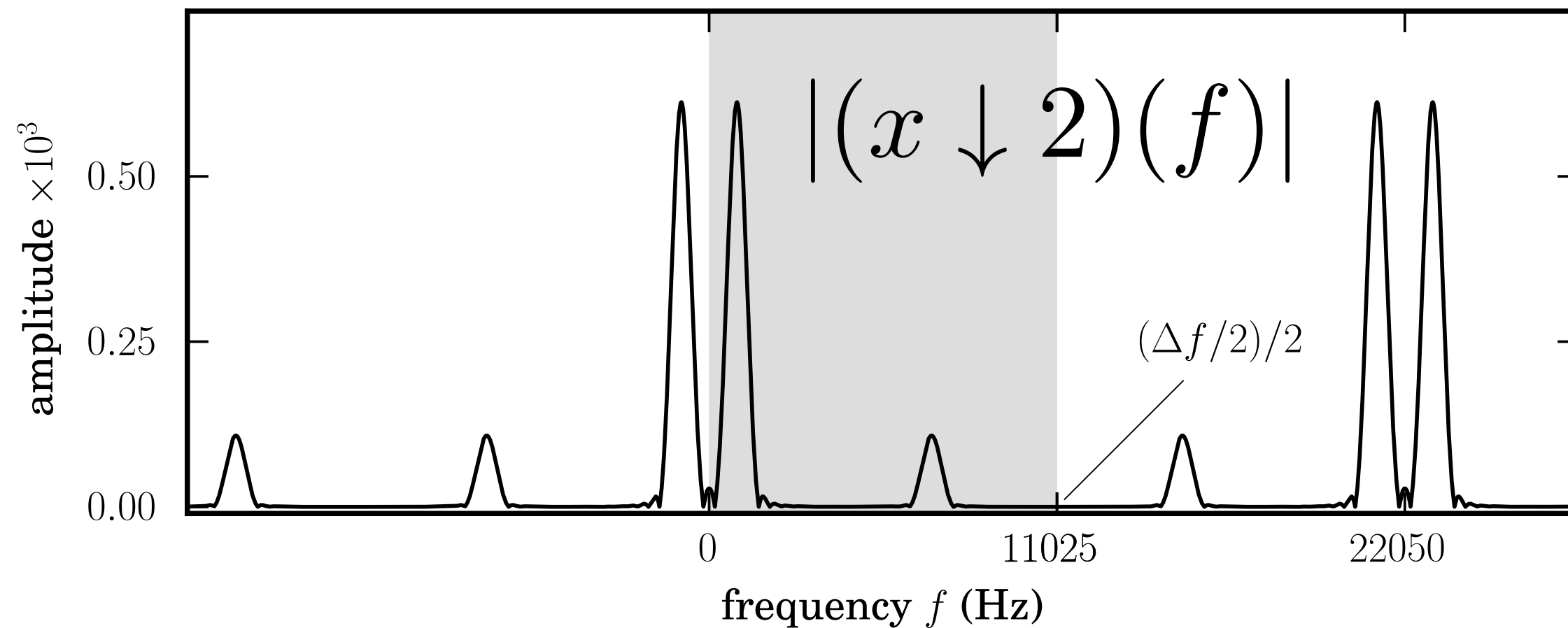
$$(x \uparrow M)(t) = x(t), \text{ if } t \in \mathbb{Z}\Delta t, \text{ 0 otherwise.}$$

$$\Delta f = 22.05 \text{ kHz}$$



Expansion - Spectrum

$$(x \uparrow M)(f) = \frac{1}{M} x(f)$$



Upsampling

$$f_c = \frac{\Delta f}{2M}$$

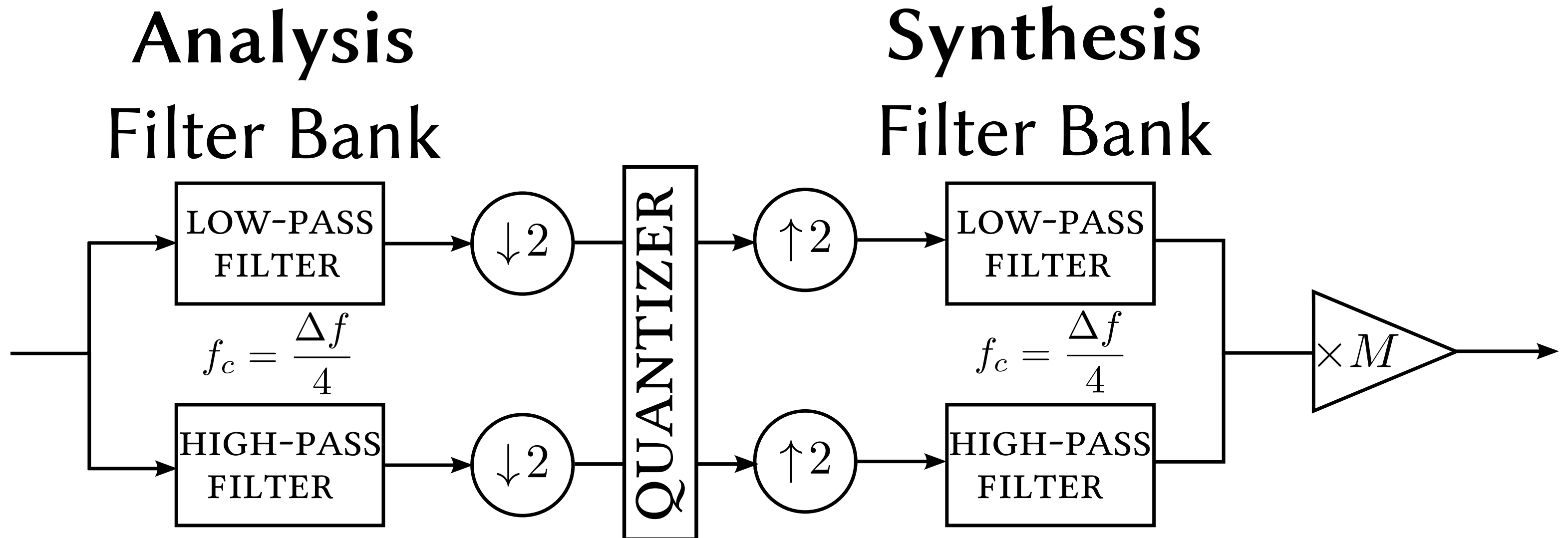


Downsampling

$$f_c = \frac{\Delta f}{2M}$$

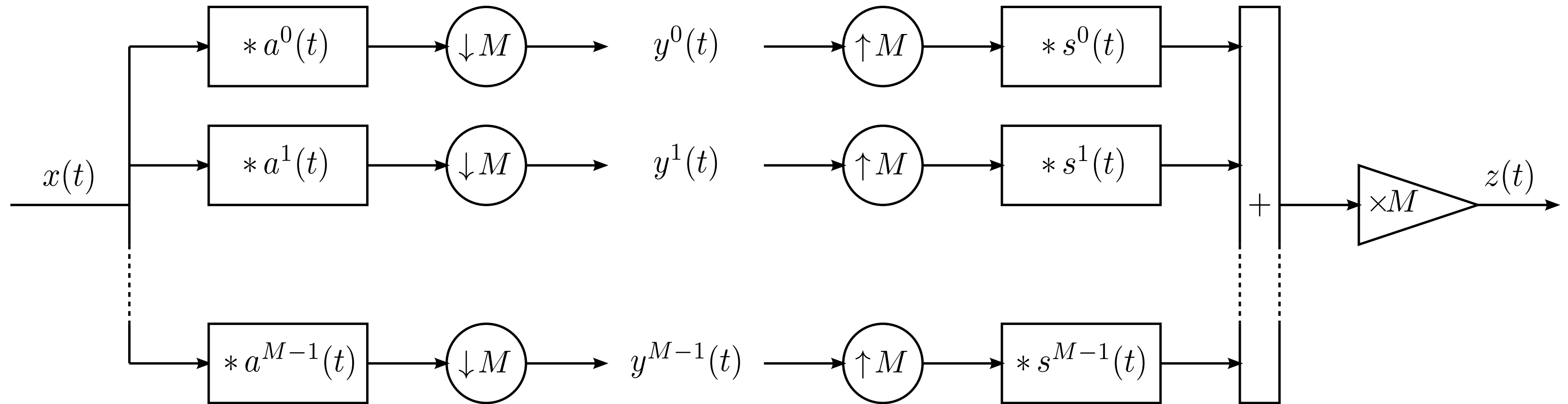


Perfect Reconstruction



If the filters are perfect and there is no quantization,
the reconstruction is perfect.

Filter Banks



$$z(f) = \sum_{k=0}^{M-1} \left[\sum_{i=0}^{M-1} s_i(f) a_i(f + k\Delta f / M) \right] x(f + k\Delta f / M)$$

Distortion

The filter banks output satisfies:

$$z(f) = x(f) + \sum_{k=0}^{M-1} D_k(f) x(f + k\Delta f/M)$$

where the k -th distortion function D_k is

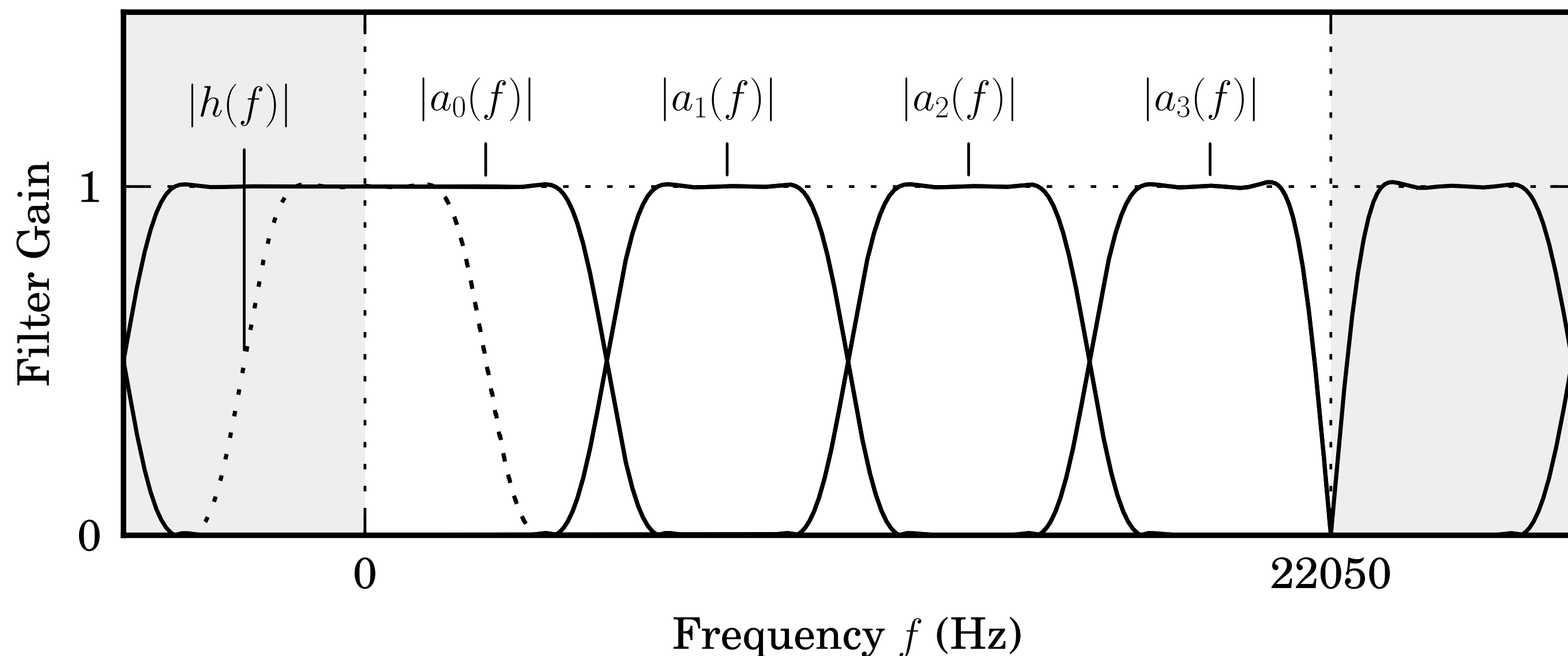
$$D_k(f) = \sum_{i=0}^{M-1} s_i(f) a_i(f + k\Delta f/M) - \delta_k$$

Perfect reconstruction:

$$\forall k \in \{0, \dots, M-1\}, \forall f \in \mathbb{R}, D_k(f) = 0$$

Modulated Filter Banks

Pick a lowpass filter with $f_c = \Delta f / 4M$,
and a frequency response $h(f)$.



Modulate the prototype filter by

$$\Delta f_k = (k + 0.5) \times \Delta f / 2M$$

Modulated Filter Banks

Example: select a prototype truncated from

$$h(n\Delta t) = \frac{\Delta f}{2M} \text{sinc} \frac{n}{2M}$$

and as

$$\exp(i2\pi\Delta f_k \times n\Delta t) = \exp(i\pi(k + 0.5)n/M)$$

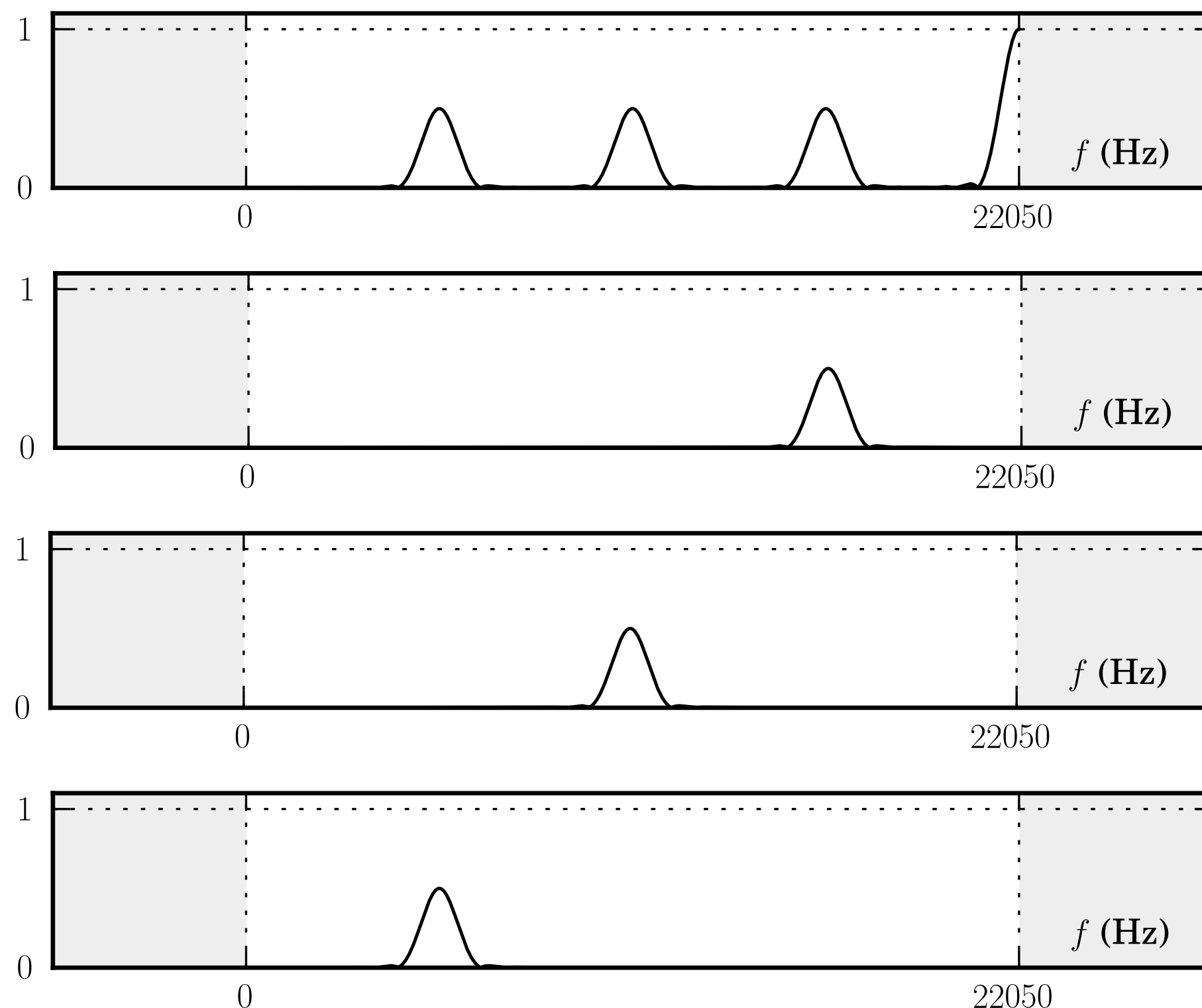
select

$$a^k(n\Delta t) = h(n\Delta t) \times 2 \cos(\pi(k + 0.5)n/M)$$

Modulated Filter Banks

Distortions

same analysis and synthesis filters



Pseudo-Quadrature Mirror Filters

1. Use the phase in the modulation:

$$\begin{aligned}a^k(n\Delta t) &= h(n\Delta t) \times 2 \cos(\pi(k + 0.5)n/M + \phi_k) \\s^k(n\Delta t) &= h(n\Delta t) \times 2 \cos(\pi(k + 0.5)n/M - \phi_k)\end{aligned}$$

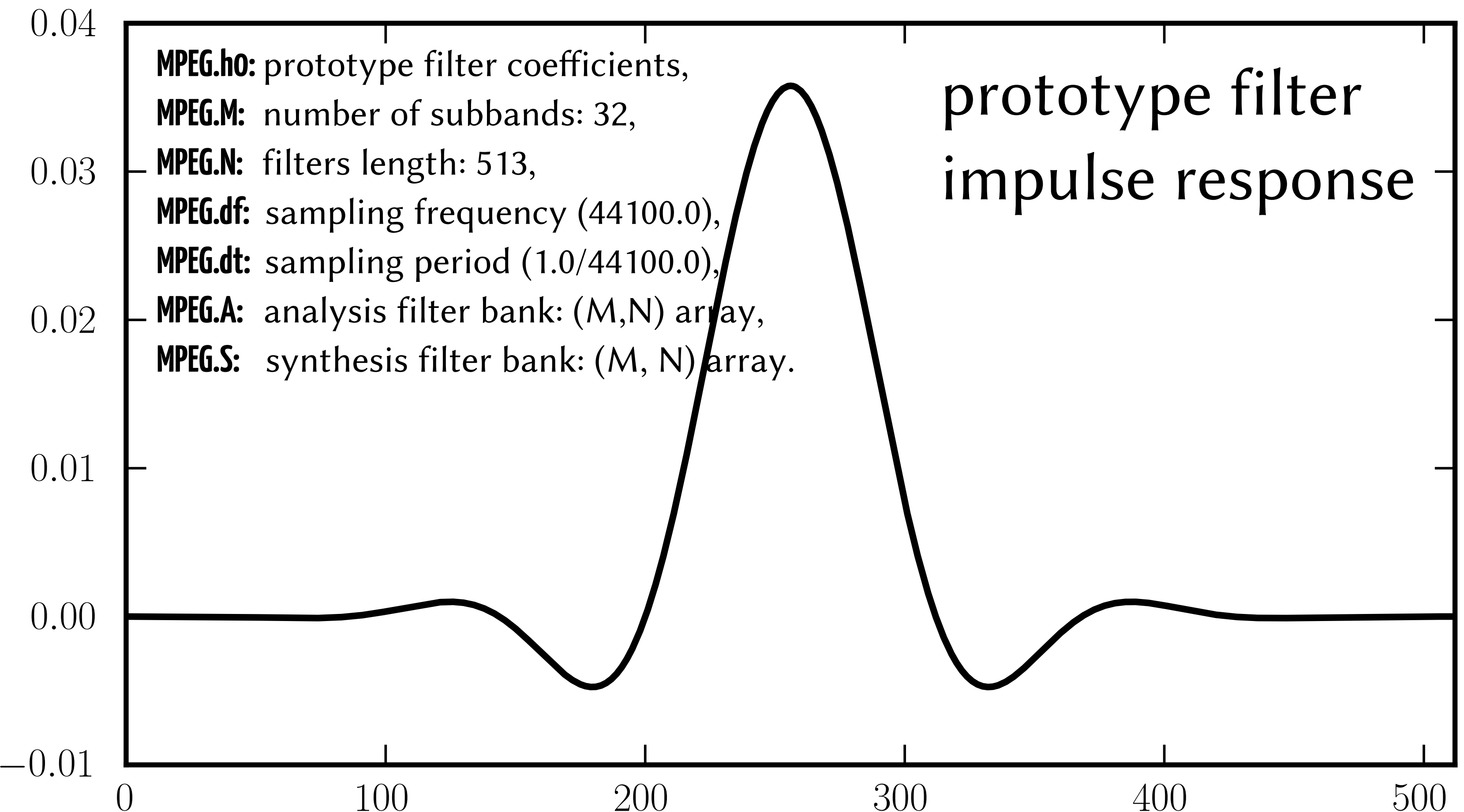
$$\phi_k = \frac{\pi}{2} \left(\frac{N-1}{M} - 1 \right) (k + 0.5)$$

where N is the prototype filter length.

2. Optimize the prototype filter w.r.t. the distortion.

Pseudo-Quadrature Mirror Filters

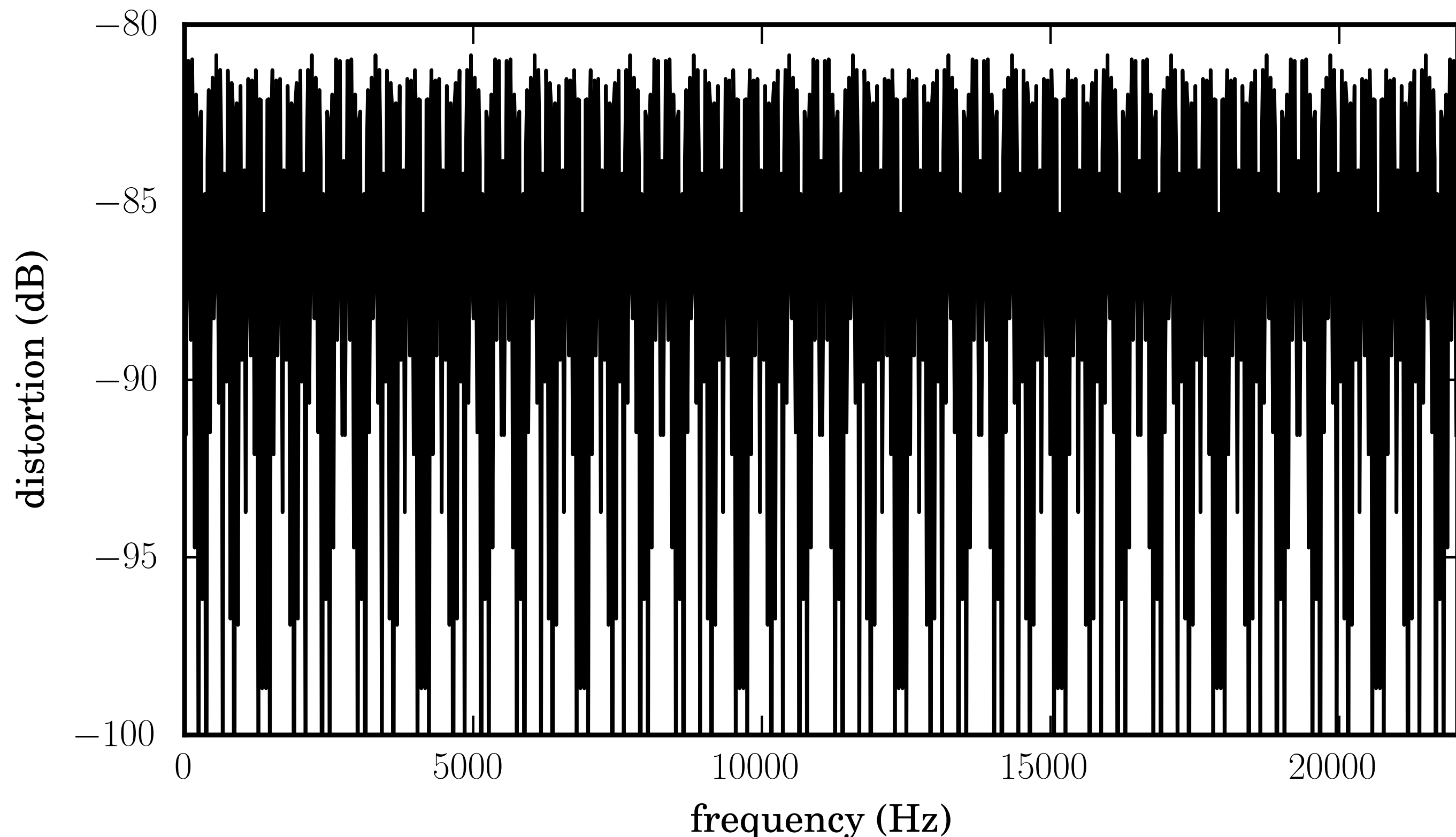
Example: MPEG Layer I + II



Pseudo-Quadrature Mirror Filters

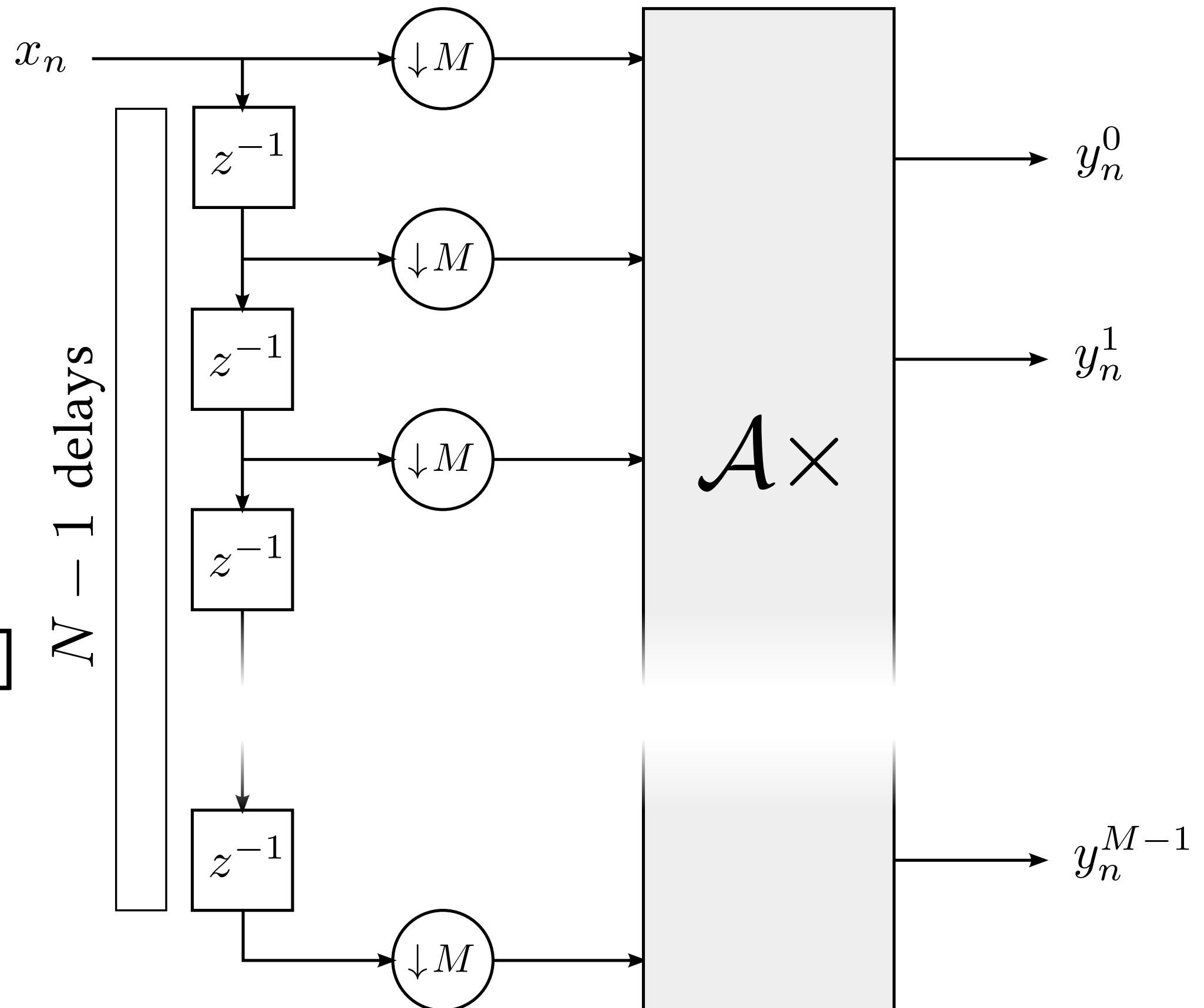
Example: MPEG Layer I + II : distortion.

$$20 \log 10 |D_0(f)|$$



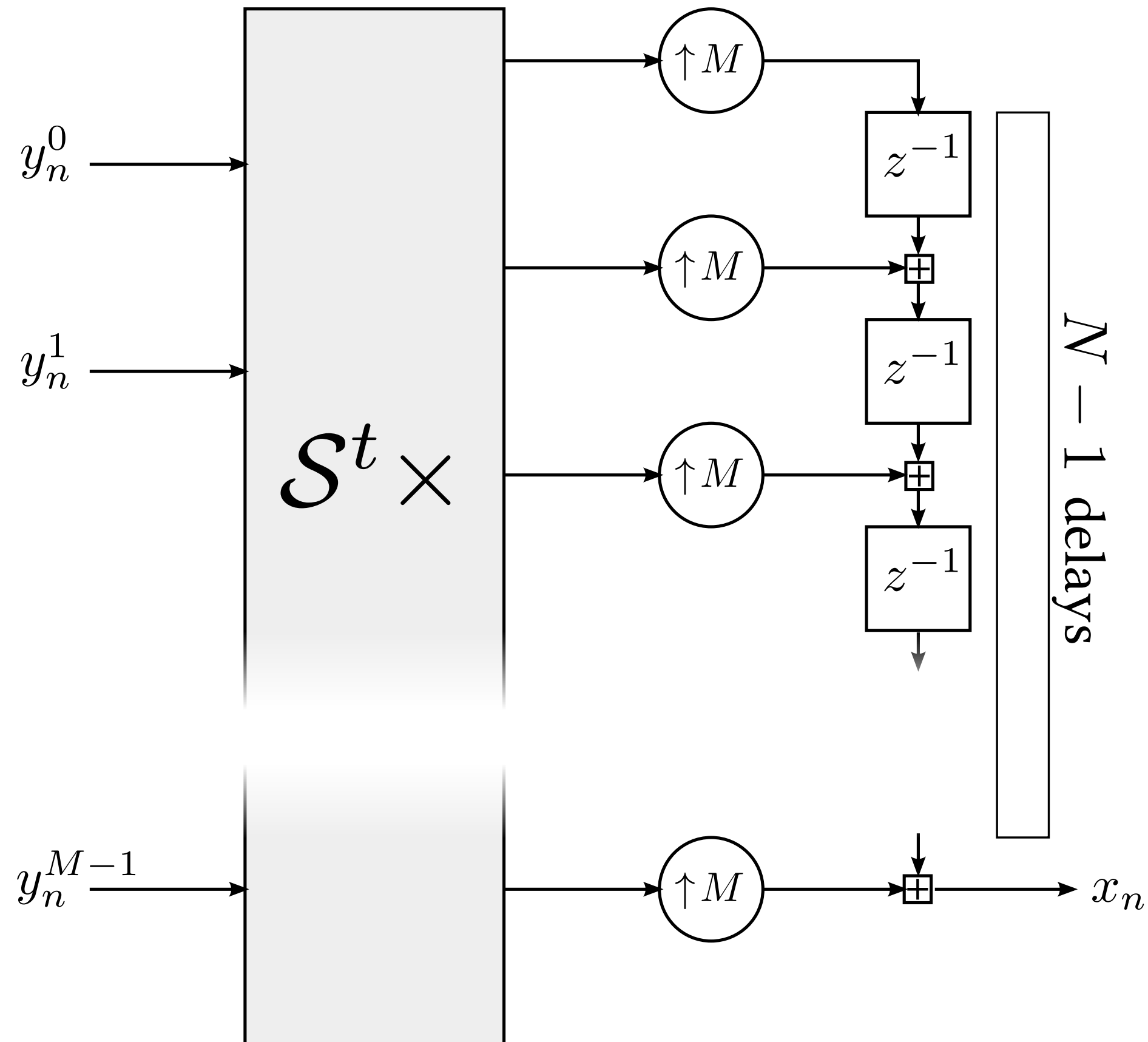
Polyphase Filter Banks

```
class Analysis(object):
    def __init__(self, a, dt=1.0):
        self.M, self.N = shape(a)
        self.A = a * dt
        self.buffer = zeros(self.N)
    def __call__(self, frame):
        self.buffer[self.M:] = buffer[:-self.M]
        self.buffer[:self.M] = frame
        return dot(self.A, self.buffer)
```



Polyphase Filter Banks

```
class Synthesis(object):
    def __init__(self, s, dt=1.0):
        self.M, self.N = shape(s)
        self.S = self.M * dt * s
        self.buffer = zeros(self.N)
    def __call__(self, frame):
        self.buffer += dot(frame, self.S)
        output = self.buffer[-self.M:].copy()
        self.buffer[self.M:] = self.buffer[:-self.M]
        self.buffer[:self.M] = zeros(self.M)
        return output
```



Acoustics

P : quadratic mean (RMS) sound pressure

$$P^2 = \langle p^2 \rangle = \frac{1}{N} \sum_{n=0}^{N-1} p(t)^2 dt$$

L : sound pressure level (dB).

$$L = 10 \log_{10} \frac{P^2}{P_0^2}, \quad P_0 = 20 \mu\text{Pa}$$

I : sound intensity

$$\frac{I}{I_0} = \frac{P^2}{P_0^2}, \quad I_0 = 10^{-12} \text{ W/m}^2$$

Acoustics

$$P^2 = \frac{2}{N\Delta t} \int_0^{\Delta f/2} |p(f)|^2 df$$

$\ell(f)$: **sound (intensity) density (dB)**

$$\ell(f) = 10 \log_{10} \frac{2}{N\Delta t} \frac{|p(f)|^2}{P_0^2}$$

$$10^{L/10} = \int_0^{\Delta f/2} 10^{\ell(f)/10} df$$

Acoustics

Normalized sound pressure waveforms

$$x(t) \in [-1.0, +1.0]$$

are interpreted as:

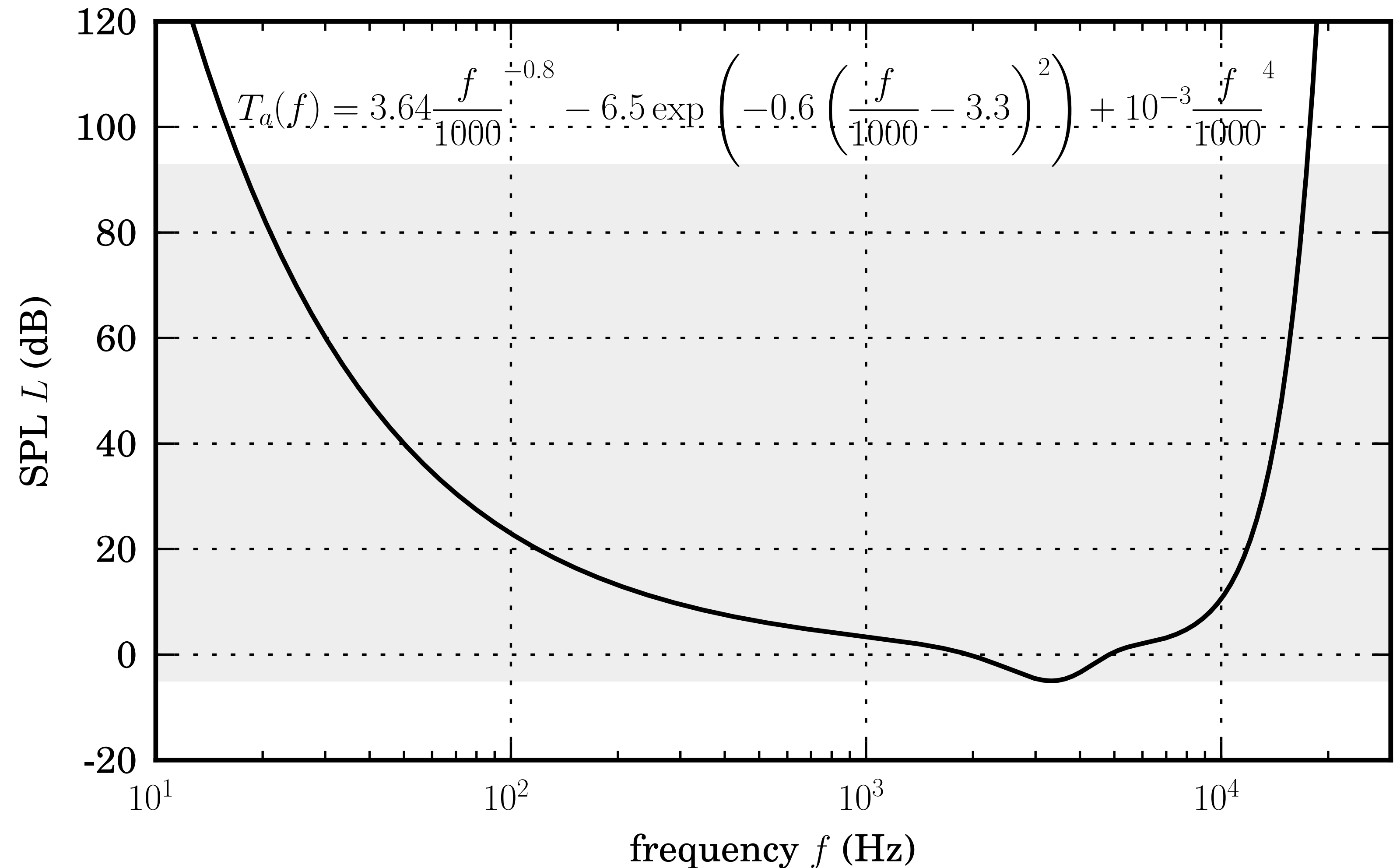
$$p(t) = 10^{4.8} P_0 \times x(t)$$

This convention yields:

$$L = 10 \log_{10} \langle x^2 \rangle + 96 \text{ dB}$$

$$\ell(f) = 10 \log_{10} \left(\frac{2}{N \Delta t} |x(f)|^2 \right) + 96 \text{ dB}$$

Absolute Threshold of Hearing



Simultaneous Masking

Fletcher's Model

Consider:

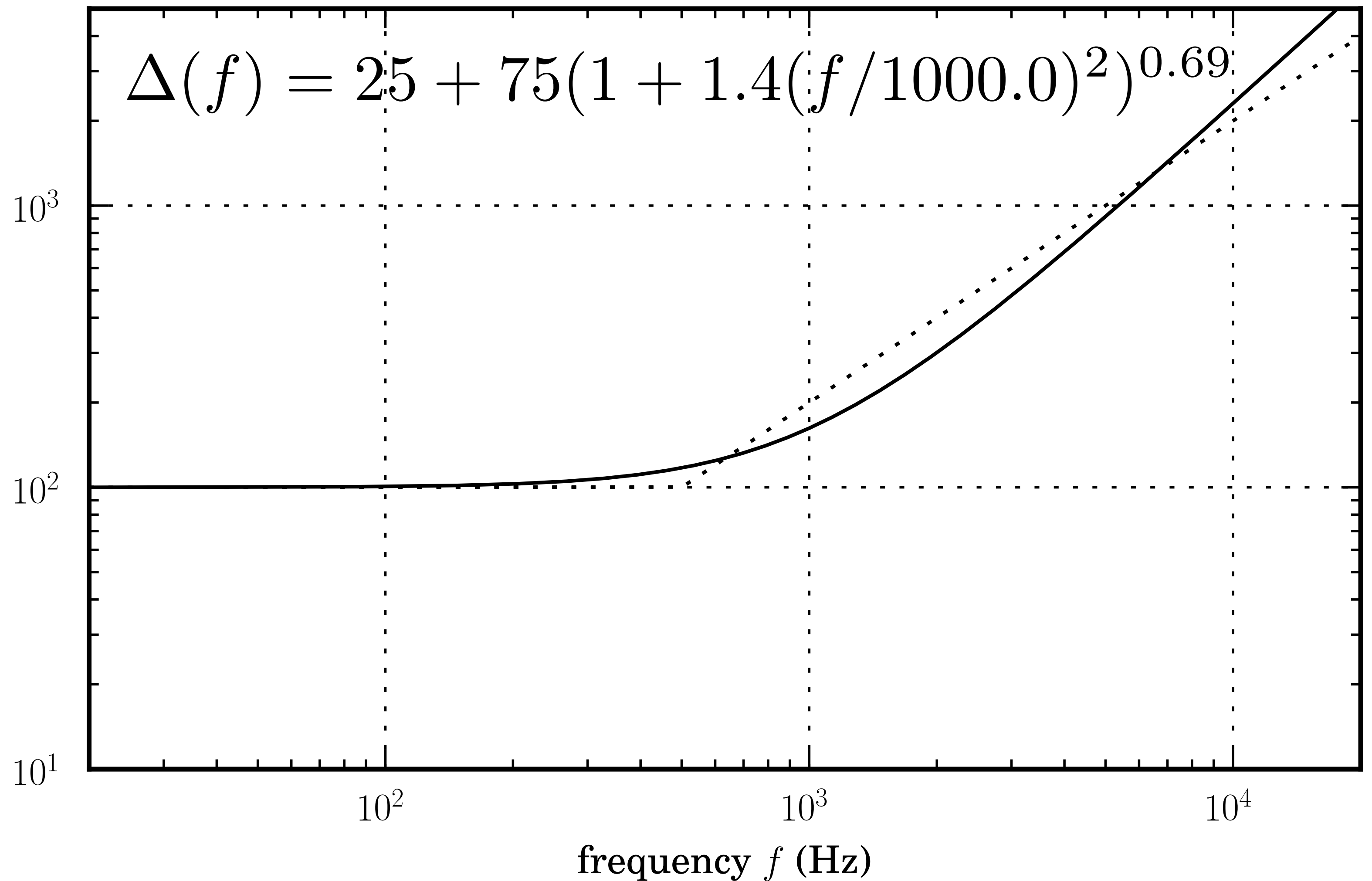
- a masker with sound density $\ell_m(f)$,
- a pure test tone with sound pressure level L and frequency f .

Masking occurs if:

$$\int_{f-\Delta(f)/2}^{f+\Delta(f)/2} 10^{\ell_m(f)/10} df \geq 10^{L/10}$$

where $\Delta(f)$ is the critical bandwidth.

Critical Bandwidth



The Bark Unit

Measure frequencies in the critical-band scale:

- first, 0 Bark corresponds to 0 Hz,
- then, +1 Bark to $+\Delta(f)$ Hz,

$$f [\text{Bark}] = 13.0 \times \arctan(0.76f/1000.0) + 3.5 \times \arctan(f/1000.0/7.5)^2$$

```
>>> from psychoacoustics import *
```

```
>>> bark([0.0, 1e4, 2e4])
```

```
array([0.0, 22.424, 24.575])
```

```
>>> hertz([0, 1, 2, 3])
```

```
array([0.0, 101.3, 203.7, 308.5])
```

```
>>> critical_bandwidth(440)
```

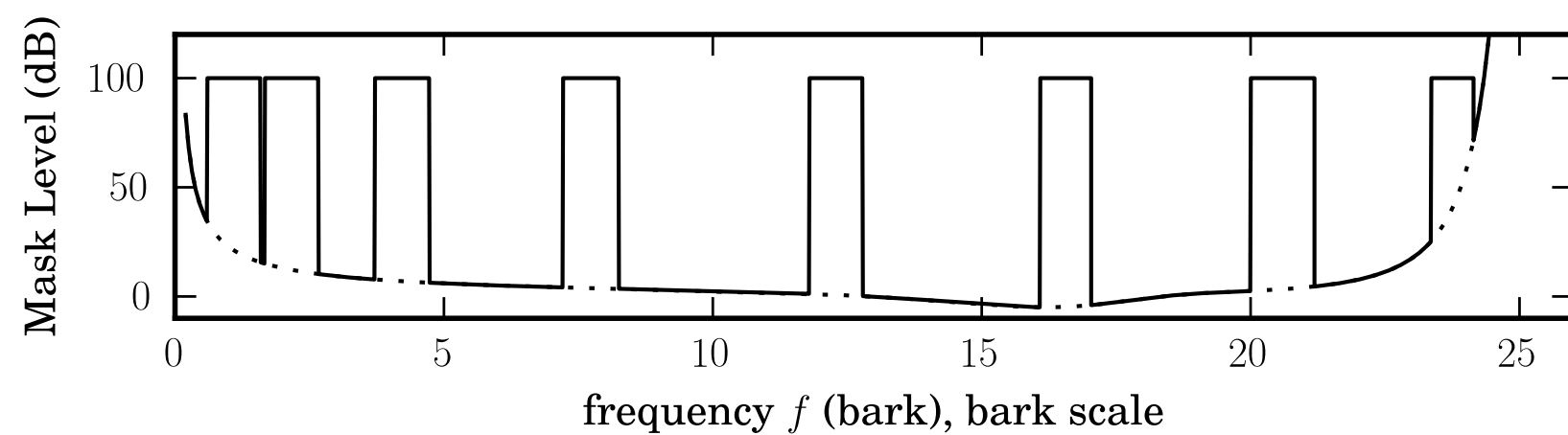
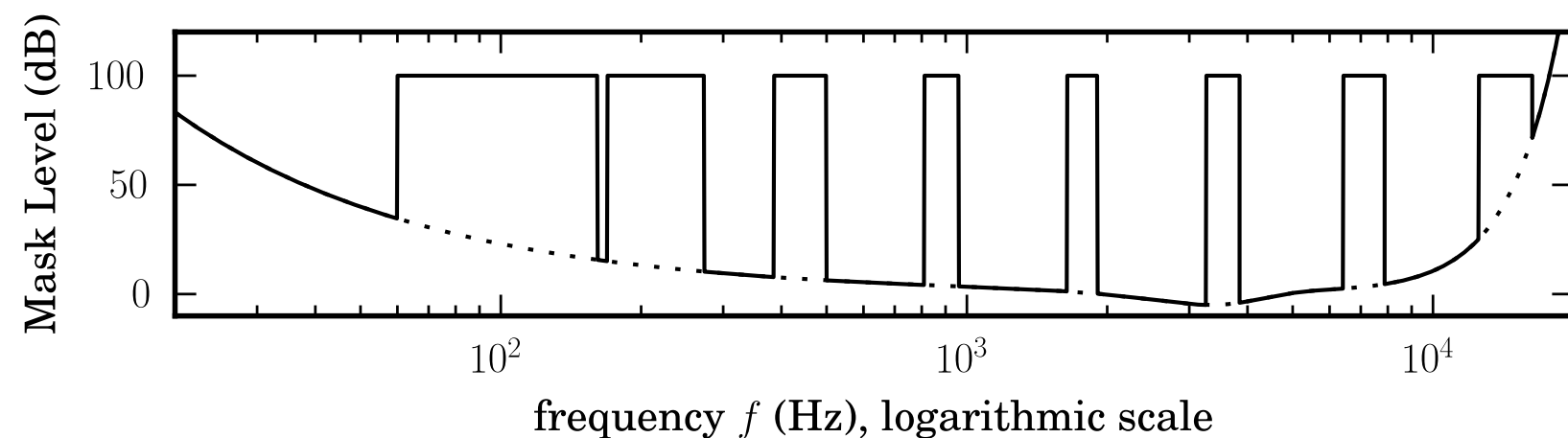
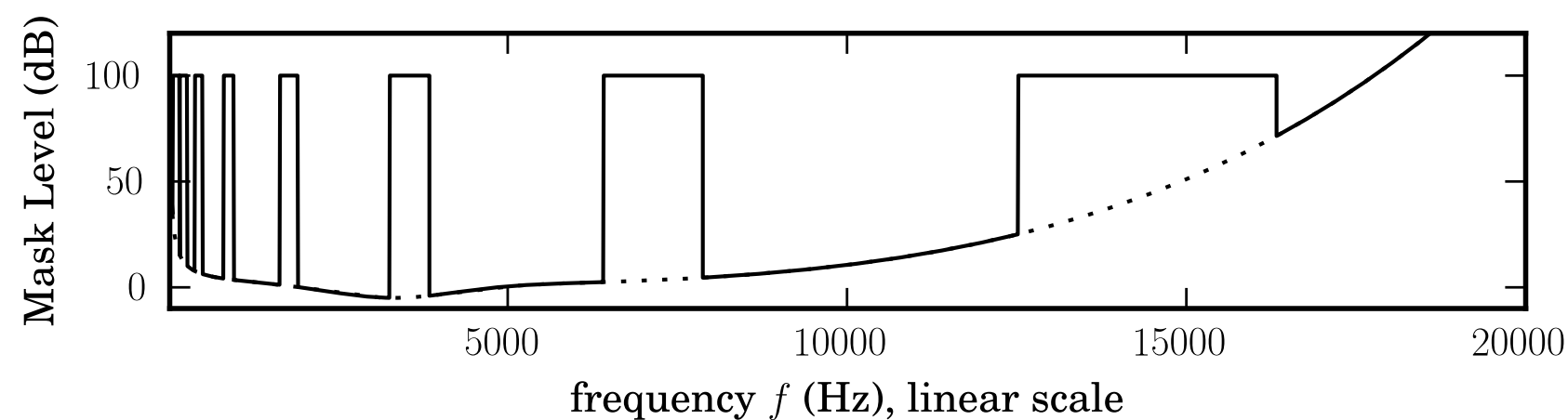
```
113.497
```

```
>>> critical_bandwidth(5000)
```

```
914.016
```

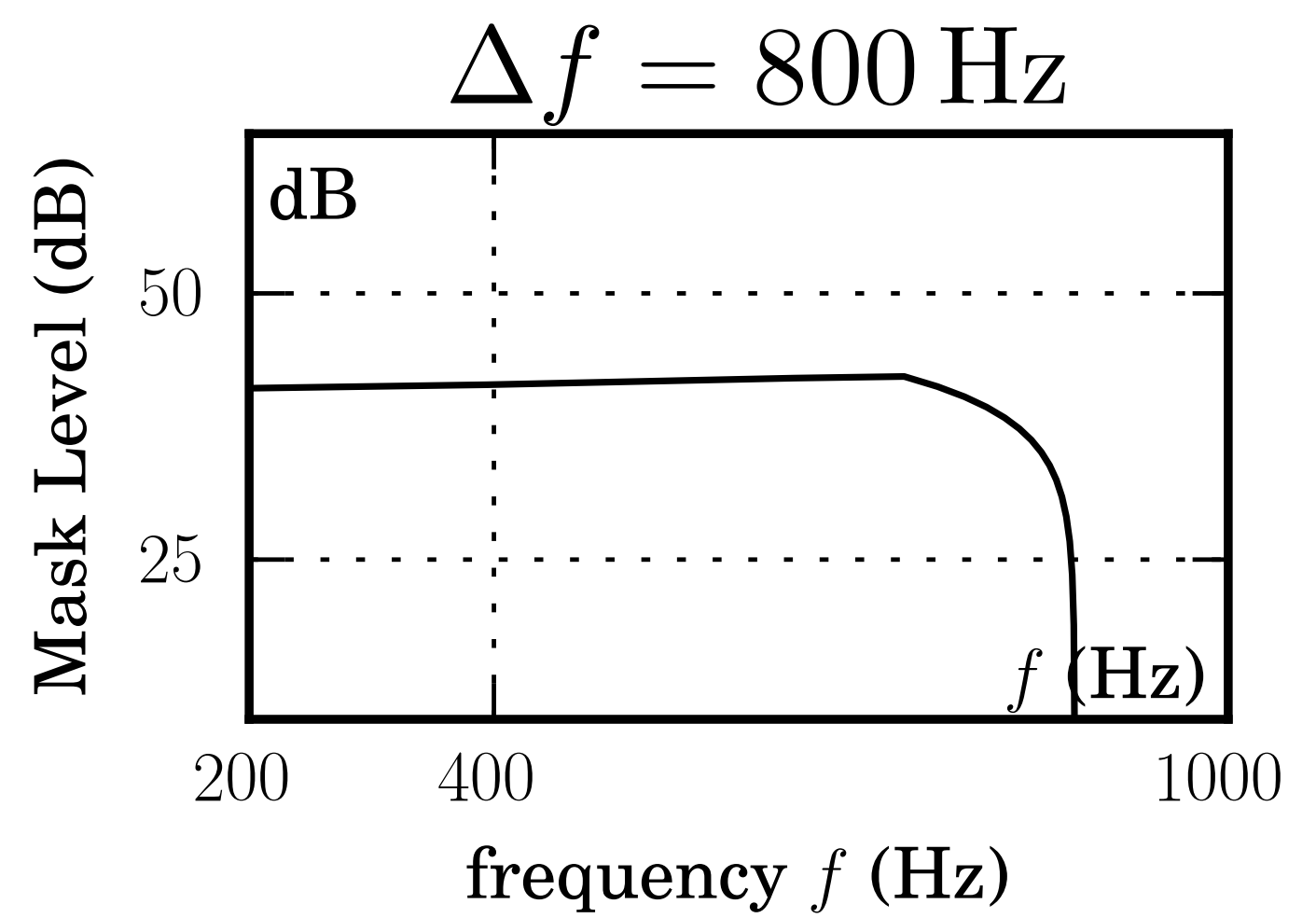
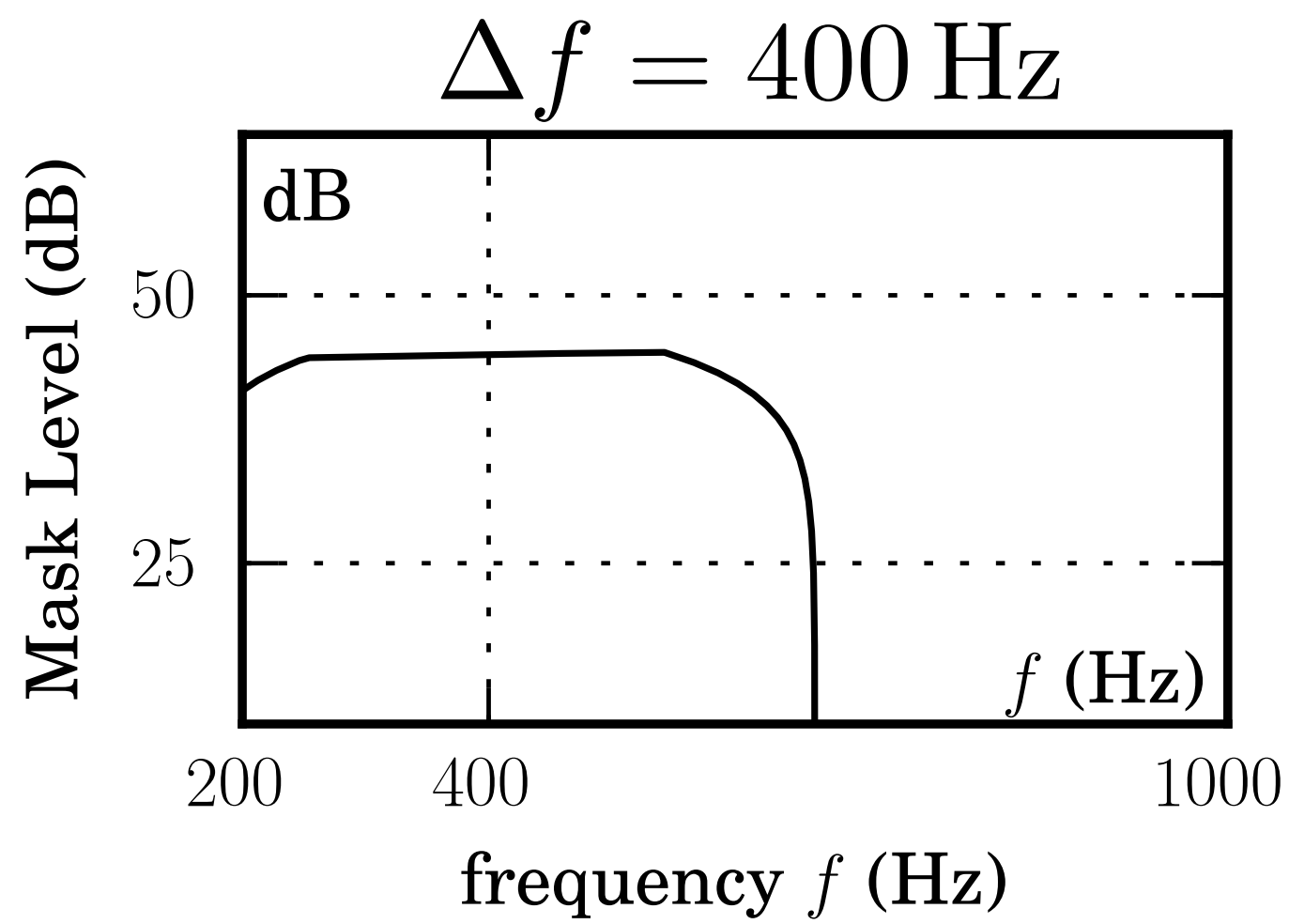
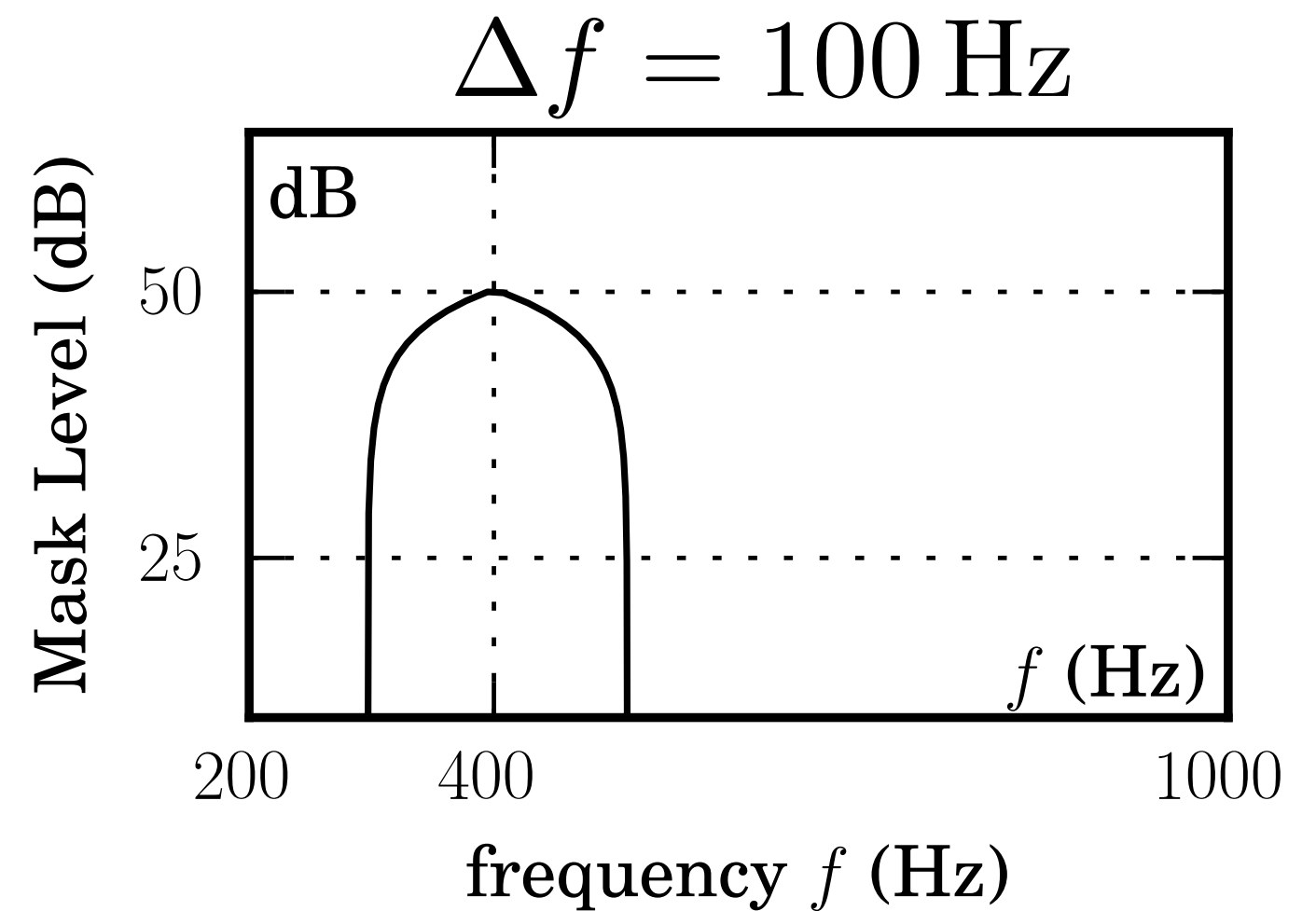
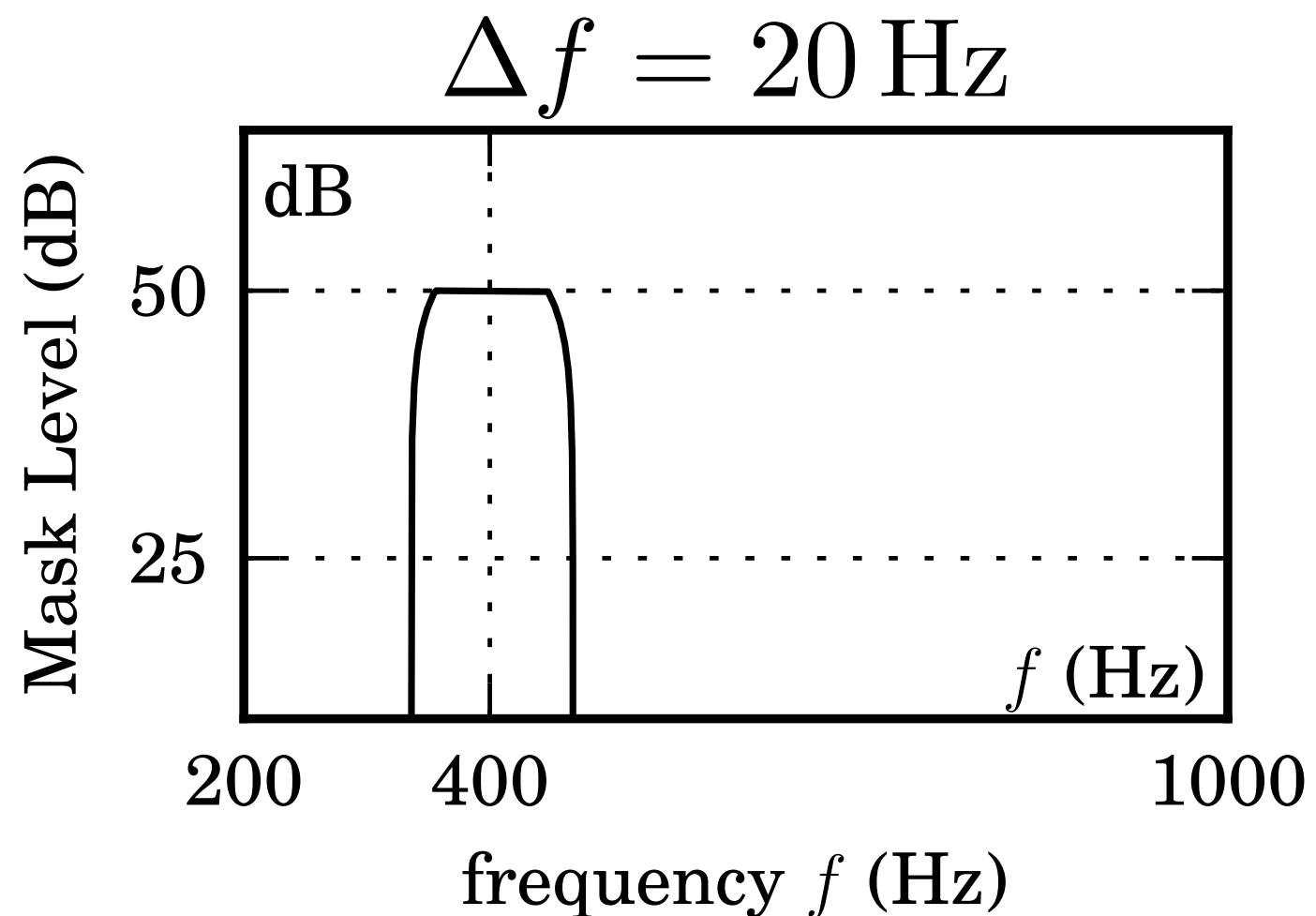
Masking by Pure Tones

Eight pure tones, with the same SPL of $L = 100$ dB.
The lowest frequency is 110 Hz, and doubles with each new masker.



Band-Limited Noise Maskers

Noise centered at 400 Hz, bandwidth Δf , $L = 50$ dB.



psychoacoustics

Mask class that acts as a compositor.

```
>>> ATH(440.0)
```

```
6.9720432781188926
```

```
>>> masker = mask(L=50, fc=400, bandwidth=0)
```

```
>>> masker(440.0)
```

```
50.0
```

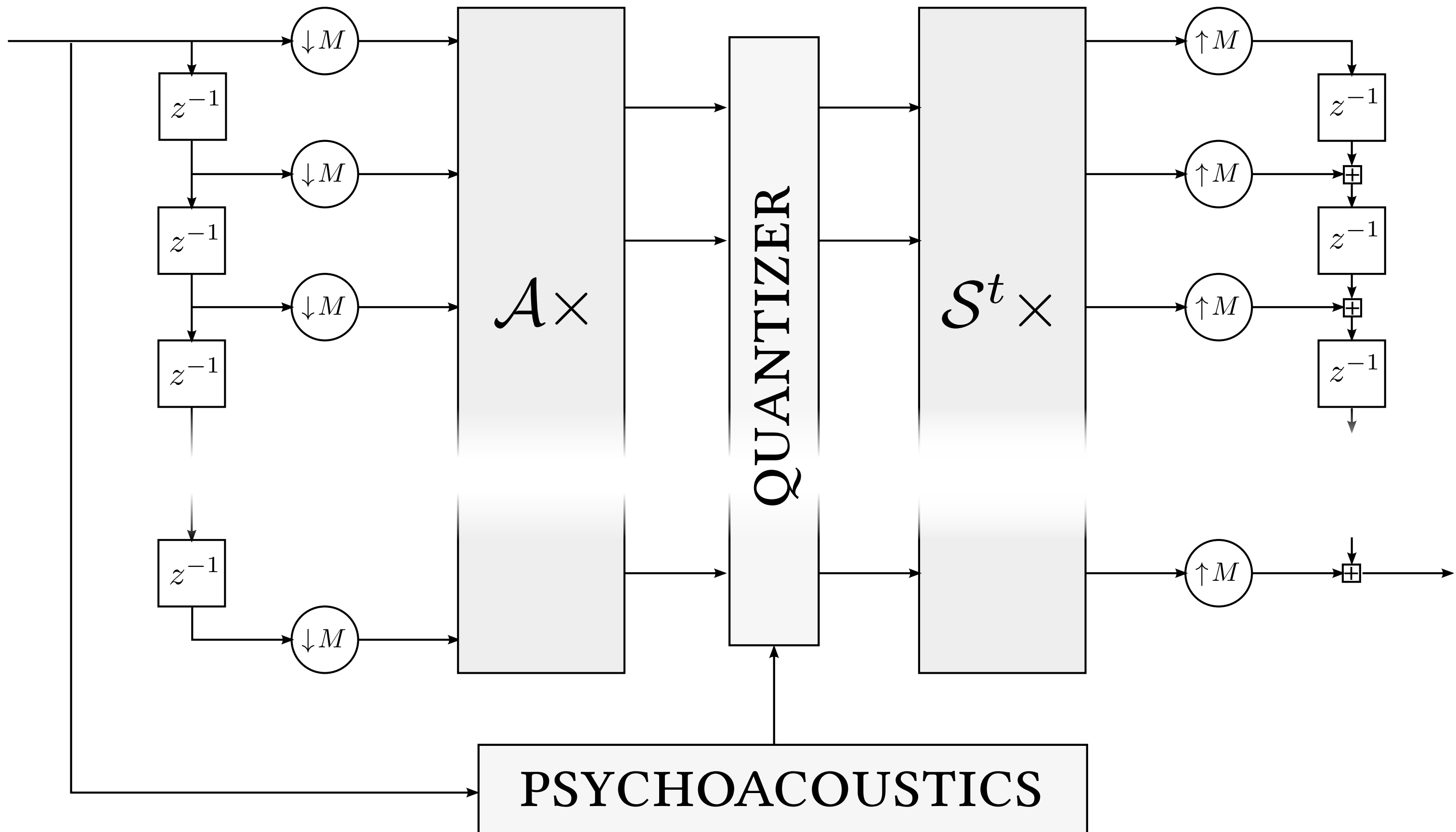
```
>>> (masker + ATH)(440)
```

```
50.00021626
```

sum of intensities



Filter Banks: Bit Allocation



Filter Banks: Bit Allocation

Let X_k be the signal component in subband k , $P_m(k)$ the (normalized) masking level intensity and $[\cdot]_k$ the corresponding quantizer.

The quantization noise is inaudible if:

$$\forall k, \mathbb{E}[(X_k - [X_k]_k)^2] \leq P_m(k)$$

Meeting these conditions may require high and/or variable bit rates, so we solve instead:

$$\min \sum_{k=0}^{M-1} \frac{\mathbb{E}[(X_k - [X_k]_k)^2]}{P_m(k)}$$

Filter Banks: Bit Allocation

If every $[\cdot]_k$ is a uniform quantizer on $(-1, +1)$ with step size Δ_k the optimal bit allocation is:

$$\Delta_k \propto \sqrt{P_m(k)}$$

If every $[\cdot]_k$ is instead optimal w.r.t. the SNR, optimal bit allocation is achieved by setting the step size of the embedded linear quantizer to:

$$\Delta_k \propto \frac{1}{\text{SMR}_k} \quad \text{with} \quad \text{SMR}_k^2 = \frac{\mathbb{E}[X_k^2]}{P_m(k)}$$

SMR_k is the **signal-to-mask ratio** in subband k .