Digital Signal Processing for Music

Part 21: Fast Convolution

Andrew Beck

Intro

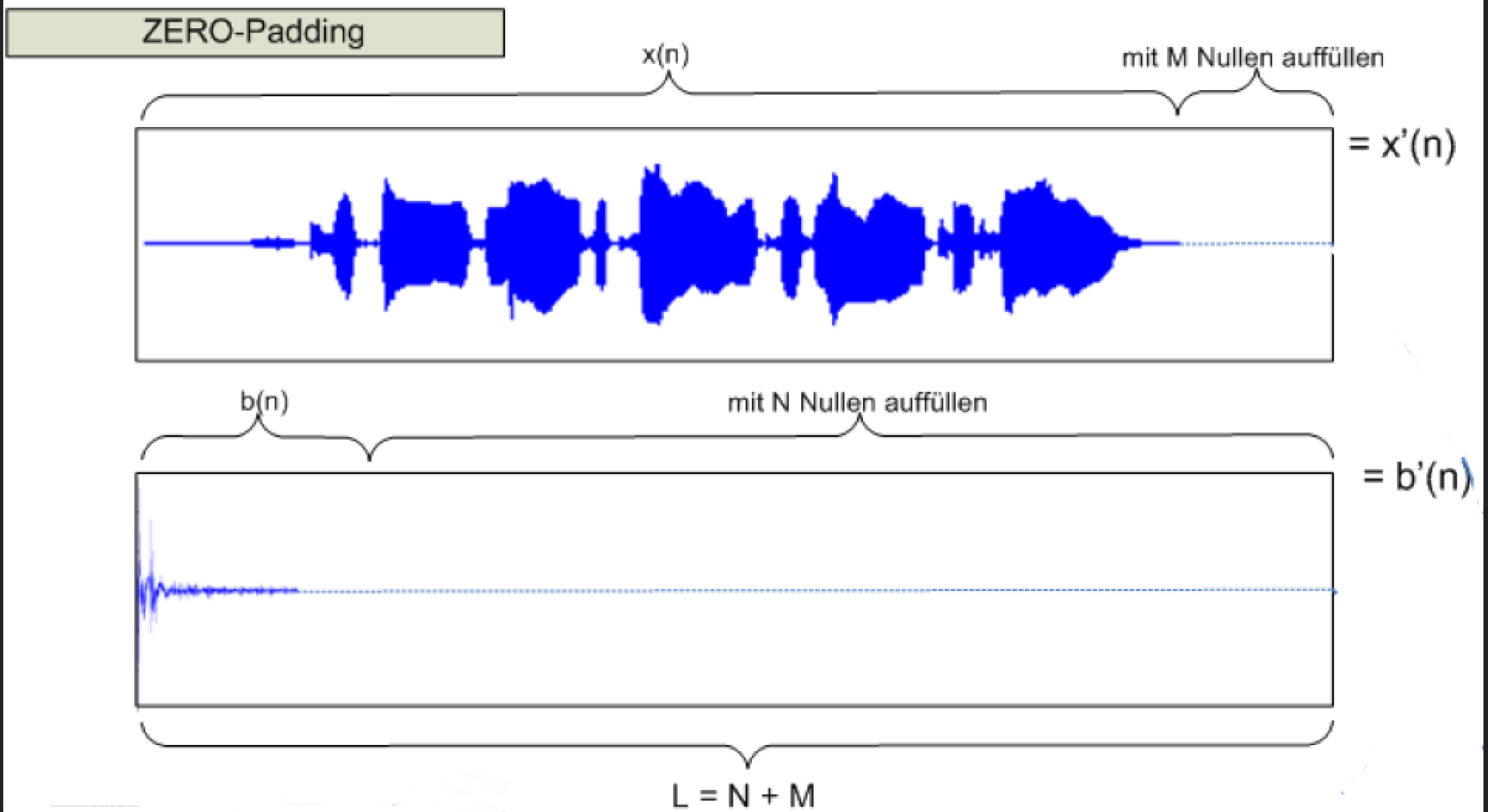
Convolution: Measure impulse response h(i) and apply FIR filter to signal

$$
\begin{aligned}
y(i) &= x(i) * h(i) \\
&= \sum_{j=-\infty}^{\infty} h(j) \cdot x(i-j)
\end{aligned}
$$

$$Y(z) = X(z) \cdot H(z)$$

Signal and Impulse Response

- Multiplication: Length of $H(z) = M$ must equal length of $X(z) = N$
- Minimum DFT length: $L \leq M + N - 1$



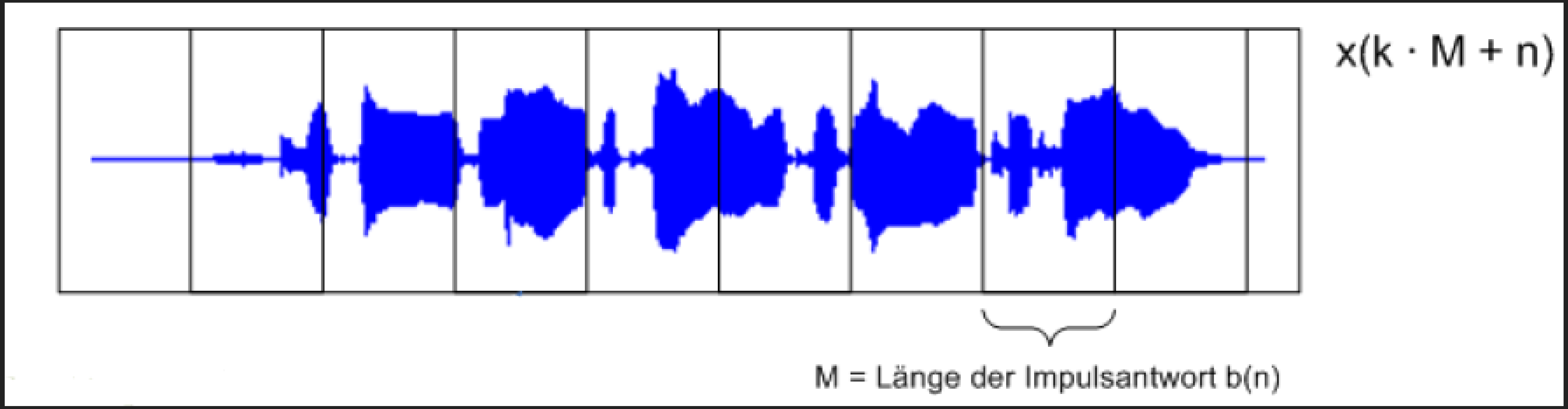
- Multiplication: Length of $H(z) = M$ must equal length of $X(z) = N$
- Minimum DFT length: $L \leq M + N - 1$

Downsides

- No real-time: Signal has to be known completely
- High memory requirements (signal length N + impulse response length M)
  - When FFT: Next larger power of two

Blocked Input Convolution

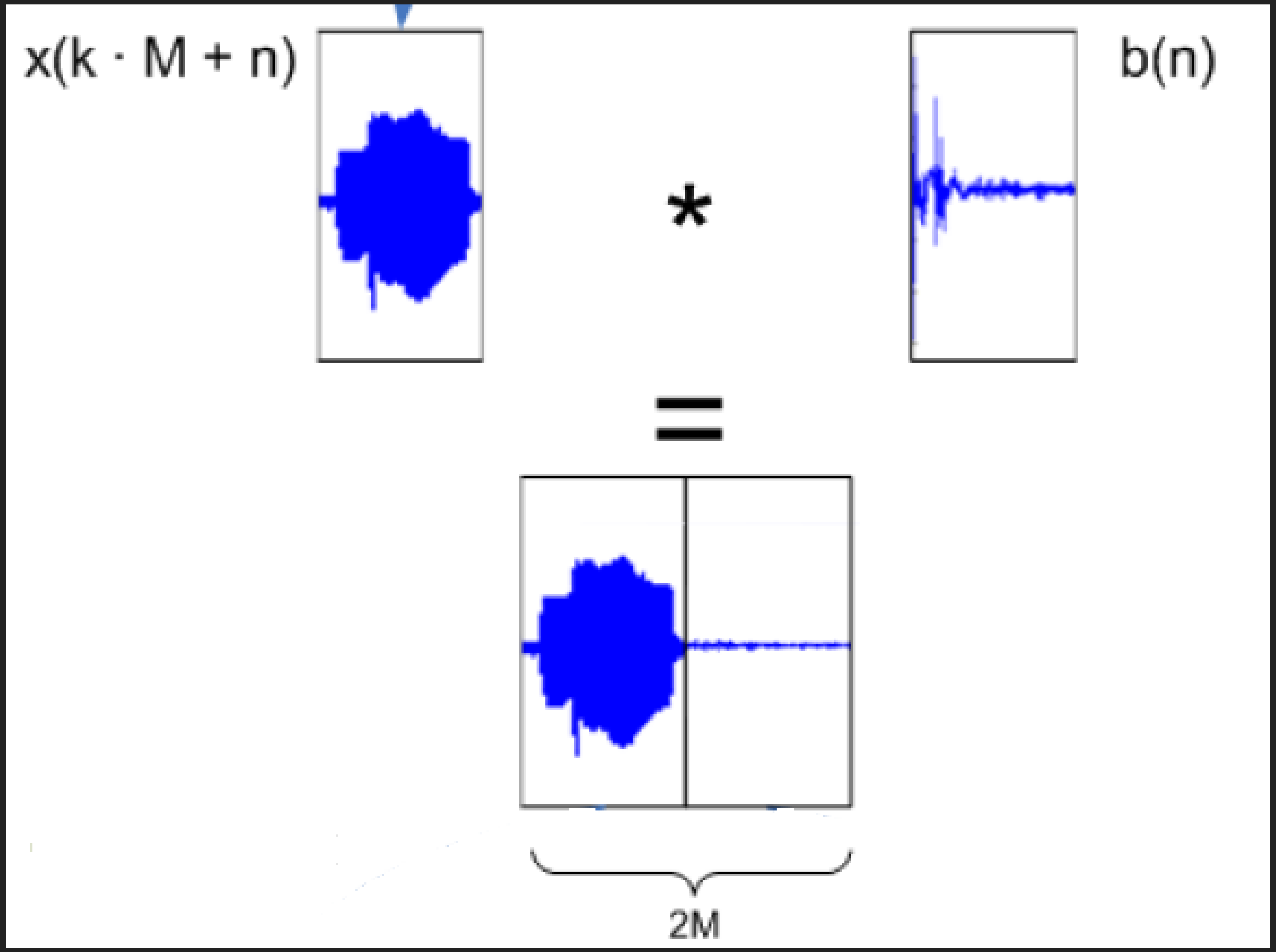
1. Split input signal into blocks of length M
2. DFT convolution with each block (zero padding)
3. Overlap and save



☰

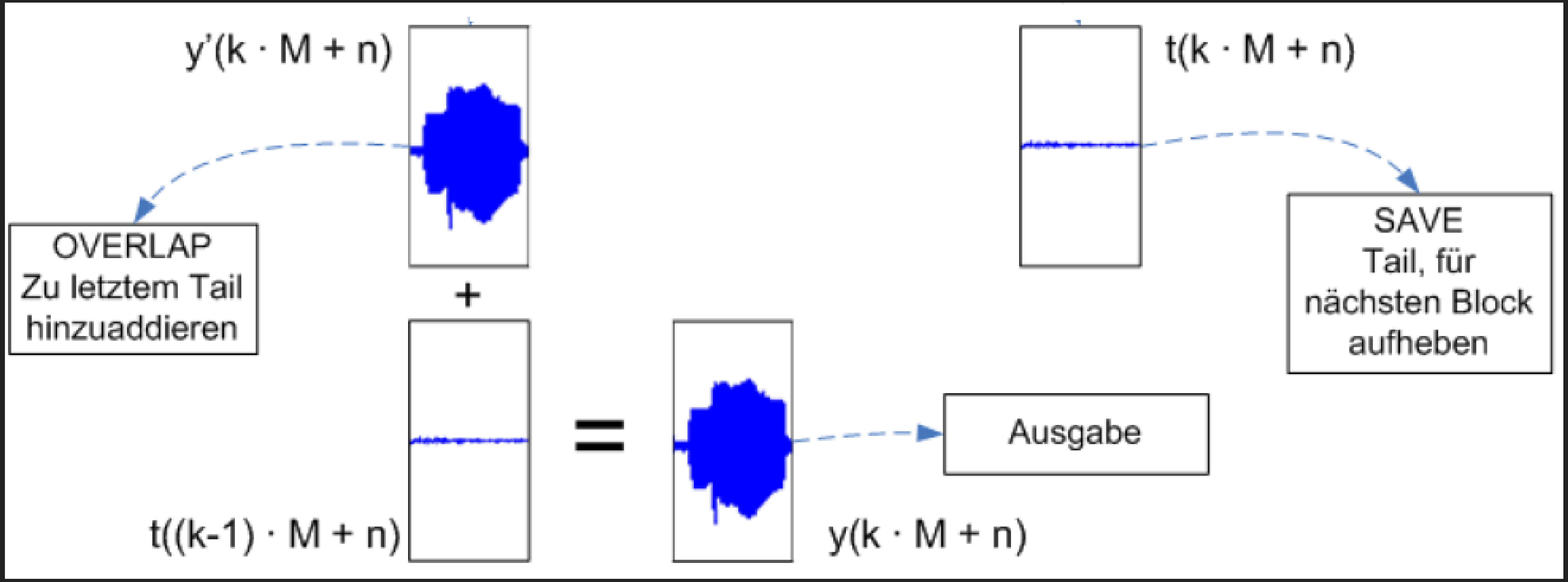
Blocked Input Convolution

1. Split input signal into blocks of length *M*
2. DFT convolution with each block (zero padding)
3. Overlap and save



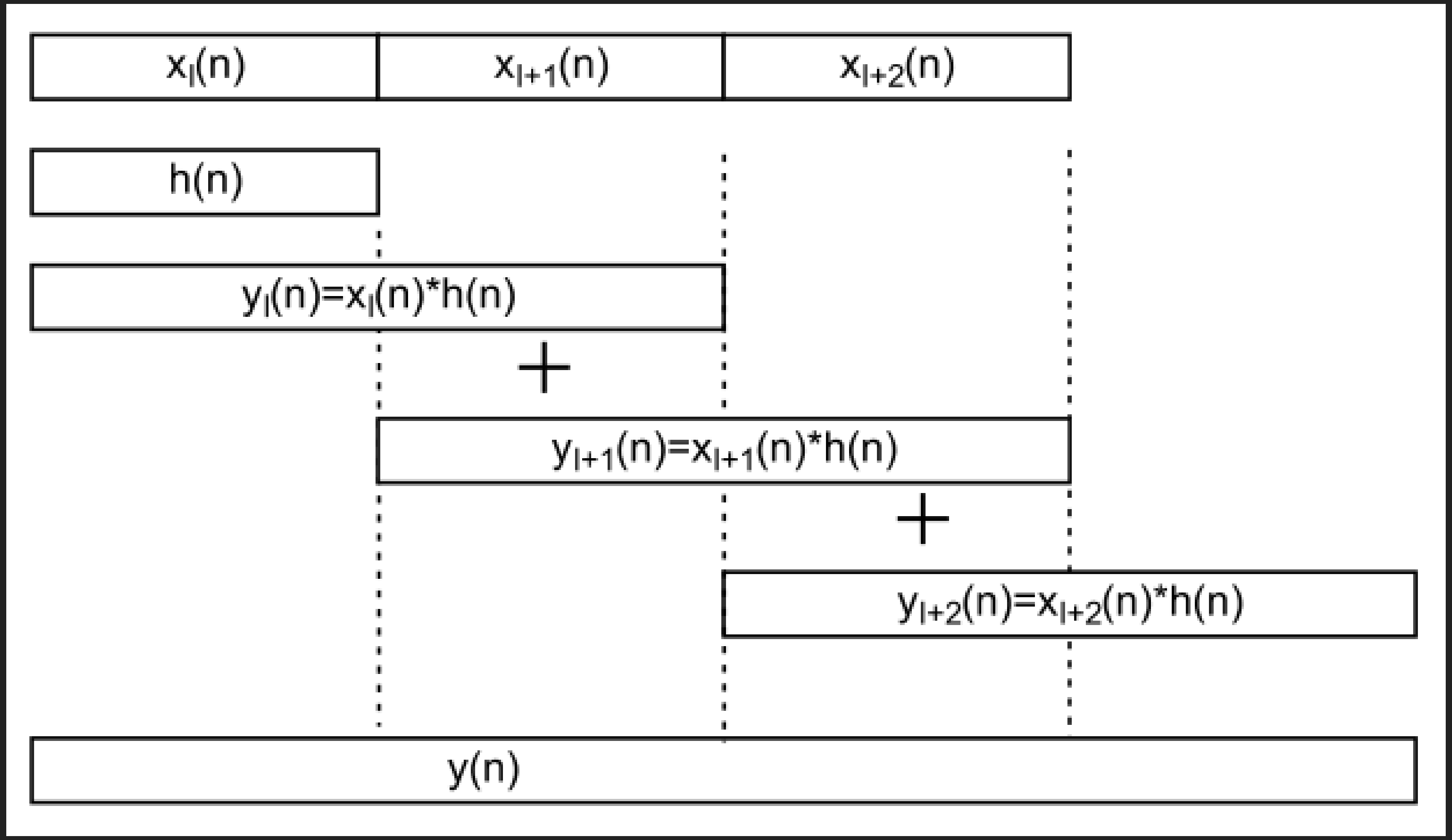
Blocked Input Convolution

1. Split input signal into blocks of length *M*
2. DFT convolution with each block (zero padding)
3. Overlap and save



Blocked Input Convolution

1. Split input signal into blocks of length M
2. DFT convolution with each block (zero padding)
3. Overlap and save

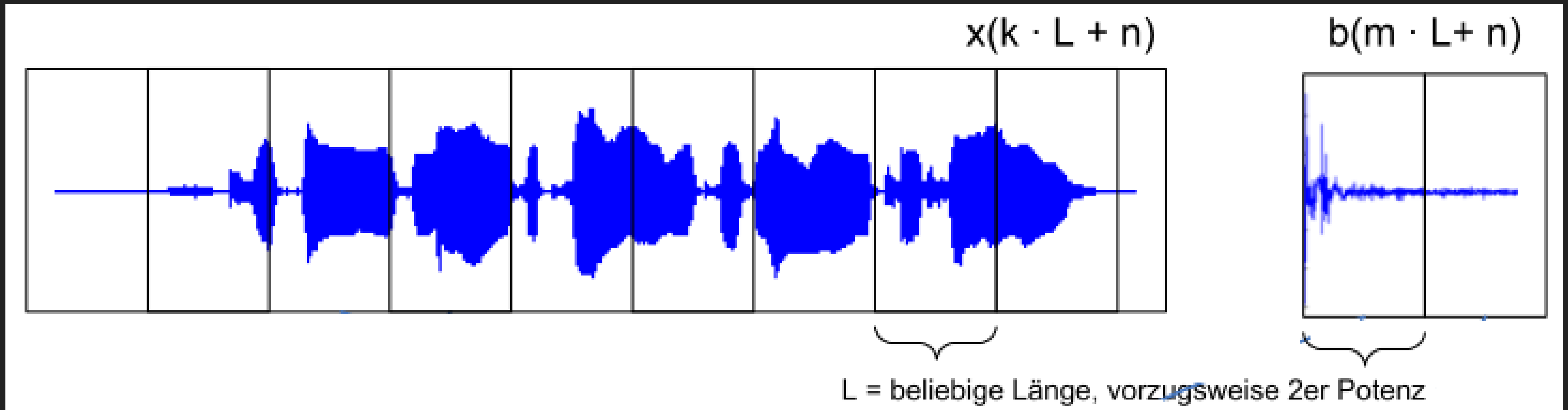


Blocked Input: Properties

- Minimum latency: Impulse response length
- Long FFT, but more efficient
- FFT of impulse response *is only computed once*

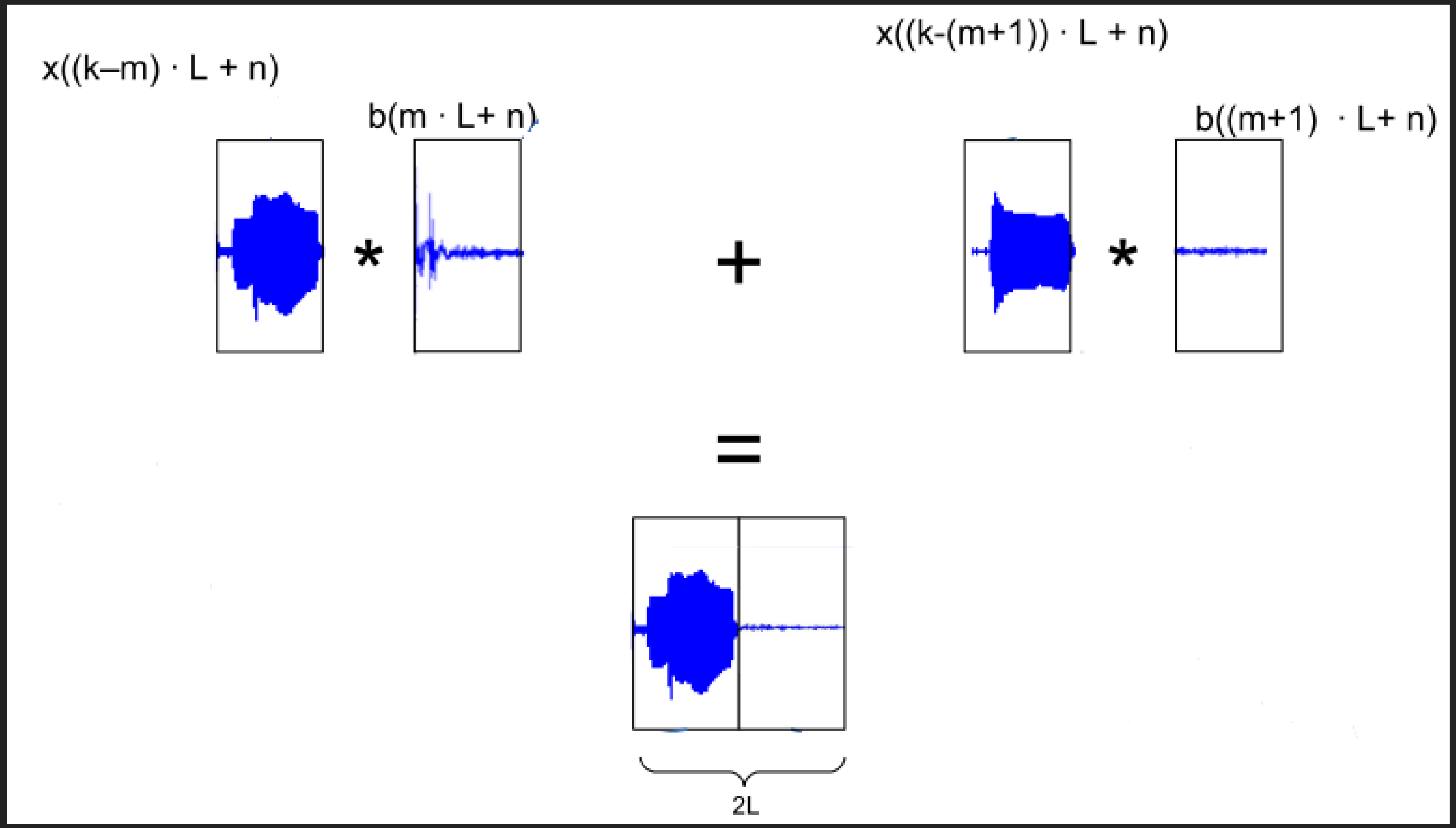
Blocked Input / Blocked Impulse Response

1. Split **both** input signal and impulse response into blocks of arbitrary length
2. DFT convolution with each signal block with each impulse response block (zero padding)
3. Overlap and save

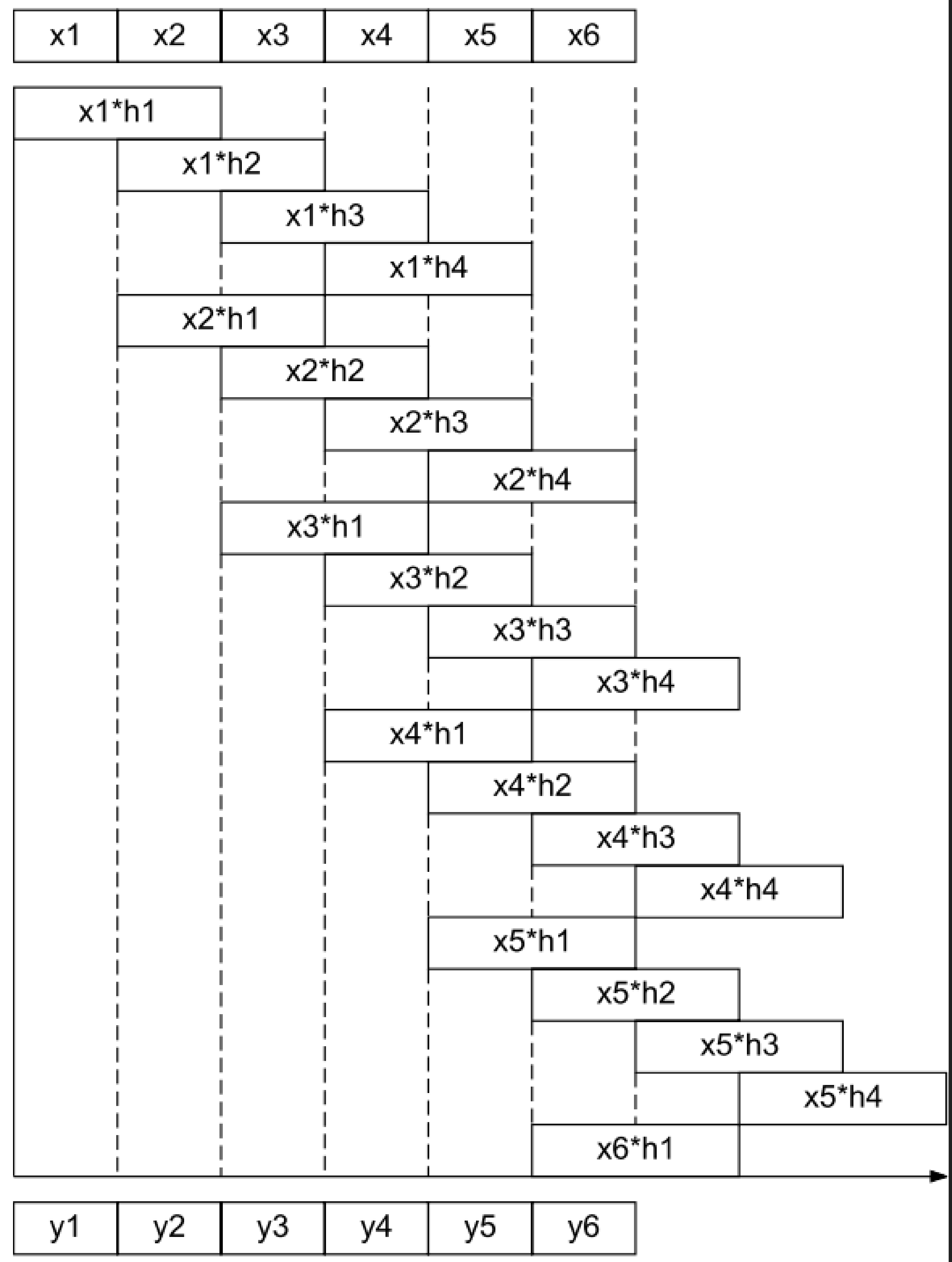


Blocked Input / Blocked Impulse Response

1. Split **both** input signal and impulse response into blocks of arbitrary length
2. DFT convolution with each signal block with each impulse response block (zero padding)
3. Overlap and save



Blocked Input / Blocked Impulse Response


x1
x2
x3
x4
x5
x6
x1*h1
x1*h2
x1*h3
x1*h4
x2*h1
x2*h2
x2*h3
x2*h4
x3*h1
x3*h2
x3*h3
x3*h4
x4*h1
x4*h2
x4*h3
x4*h4
x5*h1
x5*h2
x5*h3
x5*h4
x6*h1
y1
y2
y3
y4
y5
y6

How does block size affect processing?

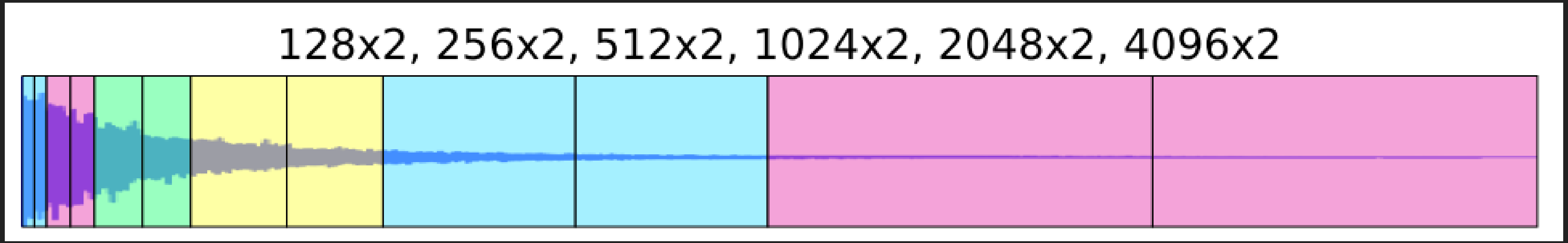
Blocked Input / Blocked Impulse Properties

- Arbitrary choice of latency / FFT length
  - Long FFT: High latency, low workload
  - Short FFT: Short latency, high workload
- FFTs of IR computed only once

≡

Variable Block Lengths

- Fast convolution: latency still formidable for efficient implementation
- Non-Uniform block lengths



- **Advantages**:
  - *any* desirable latency
- **Disadvantages**:
  - Less efficient due to multiple FFT lengths (but: inefficiency of short FFT partly compensated by very long FFTs)
  - Complex implementation
  - Comparably high memory usage (IR in many different FFT lengths