Digital Signal Processing for Music

Part 9: Fast Convolution

alexander lerch

fast convolution

introduction

convolution: measure impulse response h(i) and apply FIR filter to signal

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

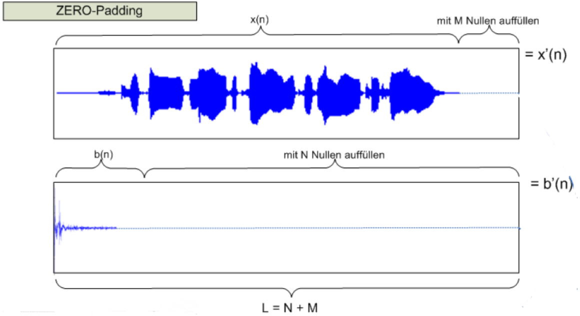
DFT convolution

signal and impulse response 1/2

- multiplication: two signals cannot be multiplied if of unequal length ($M = length(H)$, $N = length(X)$)

$\Rightarrow$ zeropad both signals

- minimum DFT length: $L \leq M + N - 1$



DFT convolution

signal and impulse response 1/2

- multiplication: two signals cannot be multiplied if of unequal length ($M = length(H)$, $N = length(X)$)

$\Rightarrow$ zeropad both signals

- minimum DFT length: $L \leq M + N - 1$

1. $X = DFT(x_\mathrm{pad}(i))$
2. $H = DFT(h_\mathrm{pad}(i))$
3. $Y = X \cdot H$
4. $y = DFT^{-1}(Y)$
5. throw away zeros if DFT was longer than $M + N$

DFT convolution

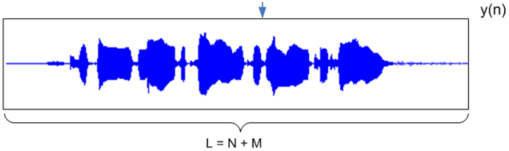
signal and impulse response 1/2

- multiplication: two signals cannot be multiplied if of unequal length ($M = length(H)$, $N = length(X)$)

$\Rightarrow$ zeropad both signals

- minimum DFT length: $L \leq M + N - 1$

1. $X = DFT(x_{\mathrm{pad}}(i))$
2. $H = DFT(h_{\mathrm{pad}}(i))$
3. $Y = X \cdot H$
4. $y = DFT^{-1}(Y)$
5. throw away zeros if DFT was longer than $M + N$



DFT convolution

signal and impulse response 2/2

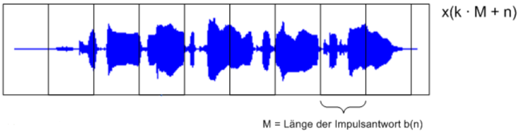
properties:

- 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 convolution

blocked signal and impulse response 1/2

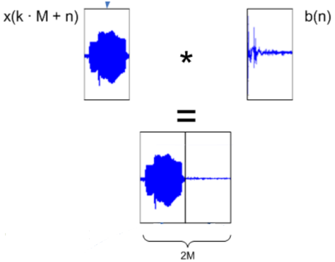
1. split input signal into blocks of length M
2. DFT convolution with each block (zeropadding)
3. overlap and save



blocked convolution

blocked signal and impulse response 1/2

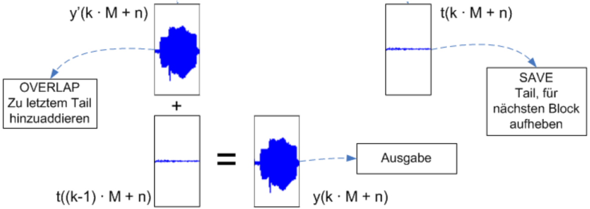
1. split input signal into blocks of length M
2. DFT convolution with each block (zeropadding)
3. overlap and save



blocked convolution

blocked signal and impulse response 1/2

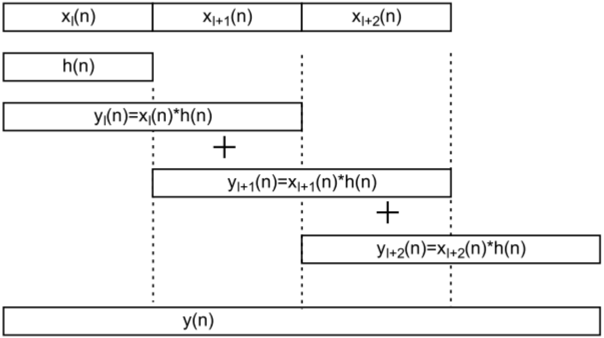
1. split input signal into blocks of length M
2. DFT convolution with each block (zeropadding)
3. overlap and save



blocked convolution

blocked signal and impulse response 1/2

1. split input signal into blocks of length M
2. DFT convolution with each block (zeropadding)
3. overlap and save



blocked convolution

blocked signal and impulse response 2/2

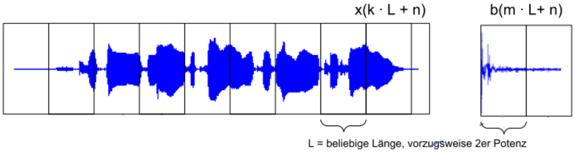
properties:

- minimum latency:
  impulse response length
- long FFT, but more efficient
- FFT of impulse response *is only computed once*

partitioned convolution

blocked signal and blocked impulse response 1/3

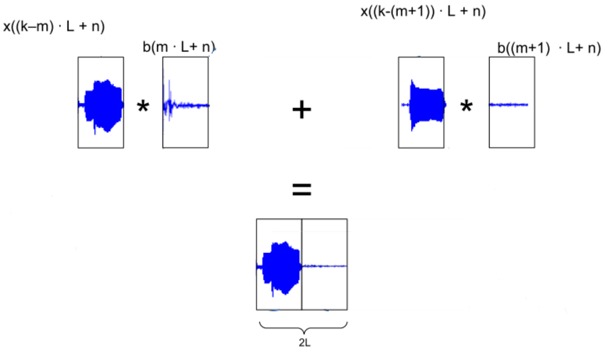
1. split **both** input signal and impulse response into blocks of arbitrary length
2. DFT convolution with each signal block with each impulse response block (zeropadding)
3. overlap and save



partitioned convolution

blocked signal and blocked impulse response 1/3

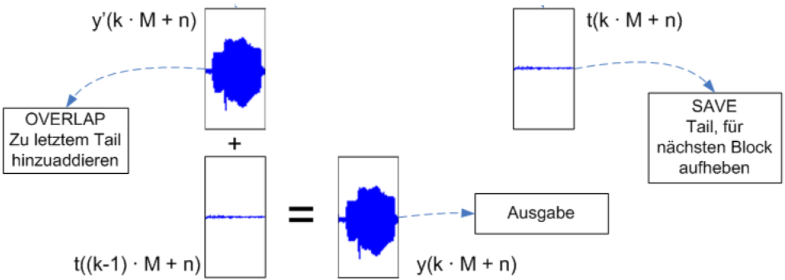
1. split **both** input signal and impulse response into blocks of arbitrary length
2. DFT convolution with each signal block with each impulse response block (zeropadding)
3. overlap and save



partitioned convolution

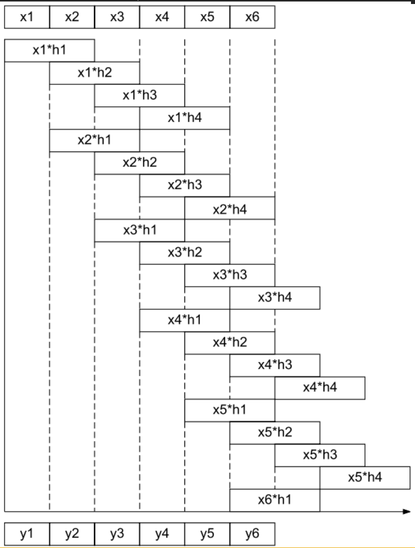
blocked signal and blocked impulse response 1/3

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 (zeropadding)
3. overlap and save



partitioned convolution

blocked signal and blocked impulse response 2/3



partitioned convolution

blocked signal and blocked impulse response 3/3

properties:

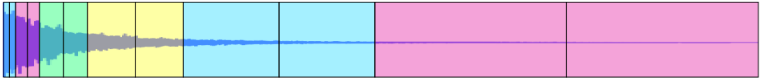
- arbitrary choice of latency/FFT length
  - long FFT: high latency, low workload
  - short FFT: short latency, high workload
- FFTs of IR computed only once

non-uniform partitioned convolution

different 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)

non-uniform partitioned convolution
different block lengths

- fast convolution: latency still formidable for efficient implementation

⇒ **non-uniform block lengths**

128x2, 256x2, 512x2, 1024x2, 2048x2, 4096x2

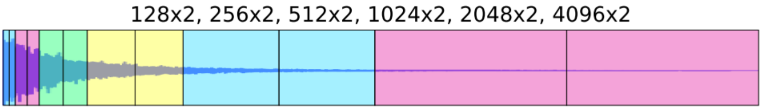
- **advantages**:
  - *any* desirable latency
- **disadvantages**:
  - less efficient due to multiple FFT lengths (but: inefficiency of short FFT partly compensated by very long FFTs)
  - complex implementation
  - comparably high memory usage (IR in many different FFT lengths)

non-uniform partitioned convolution

different 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)