COM303: Digital Signal Processing

Lecture 9: Linear Systems

Overview

- linear systems
- filtering by example
- stability

Overview

- linear systems
- filtering by example
- stability

Overview

- linear systems
- filtering by example
- stability

Overview:

- Linearity and time invariance
- Convolution

Overview:

- Linearity and time invariance
- Convolution

A generic signal processing device

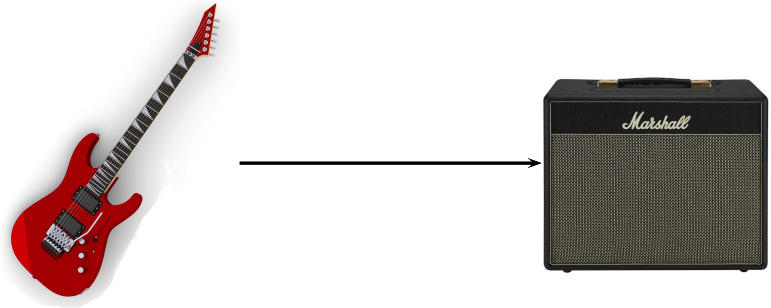
$$x[n] \longrightarrow \boxed{\mathcal{H}} \longrightarrow y[n]$$

$$y[n] = \mathcal{H}\{x[n]\}$$

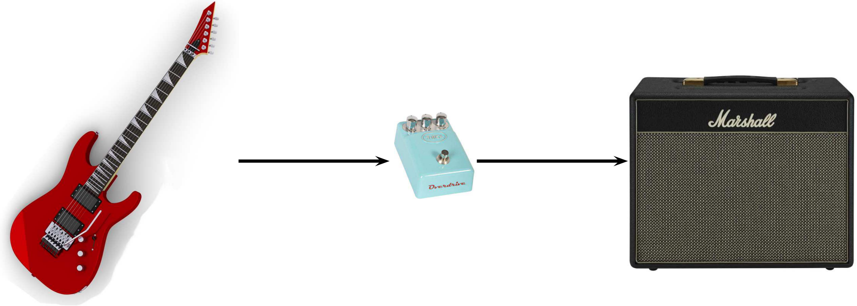
Linearity

$$\mathcal{H}\{\alpha\, x_1[n] + \beta\, x_2[n]\} = \alpha\, \mathcal{H}\{x_1[n]\} + \beta\, \mathcal{H}\{x_2[n]\}$$

Linearity



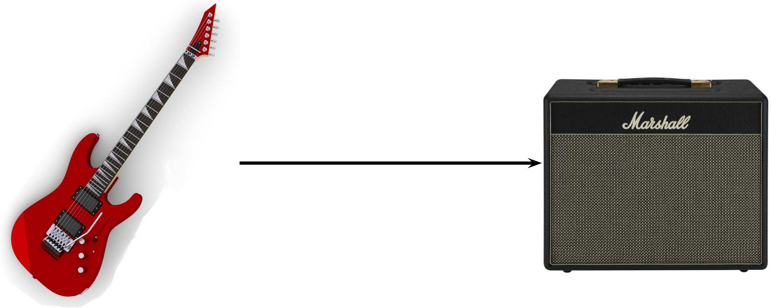
(Non) Linearity



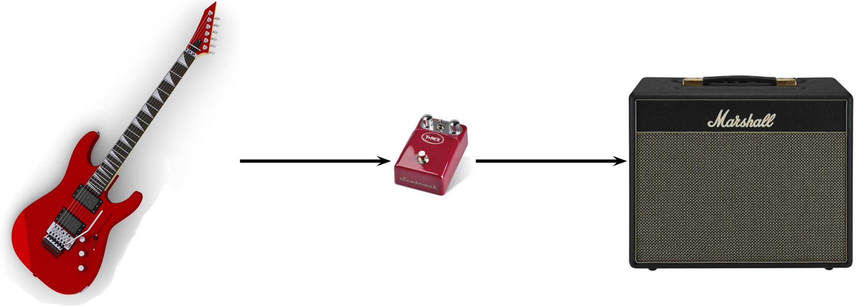
Time invariance

$$y[n] = \mathcal{H}\{x[n]\} \quad \Longleftrightarrow \quad \mathcal{H}\{x[n-n_0]\} = y[n-n_0]$$

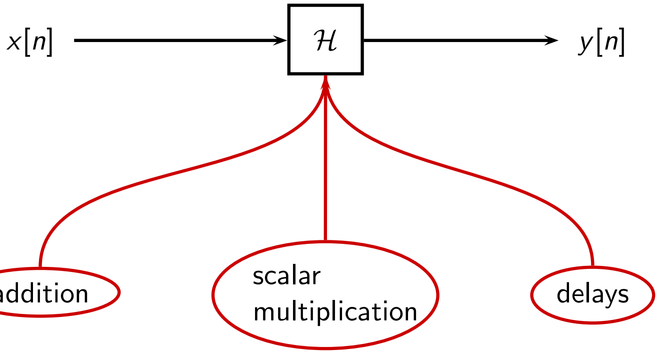
Time invariance



Time (in)variance



Linear, time-invariant systems



Linear, time-invariant systems

$$y[n] = H(x[n], x[n-1], x[n-2], \ldots, y[n-1], y[n-2], \ldots)$$

with $H(\cdot)$ a linear function of its arguments

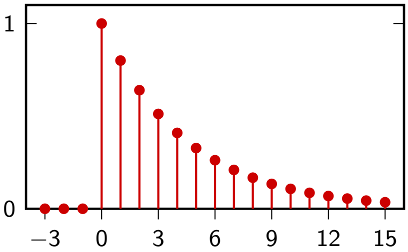
Impulse response

$$h[n] = \mathcal{H}\{\delta[n]\}$$

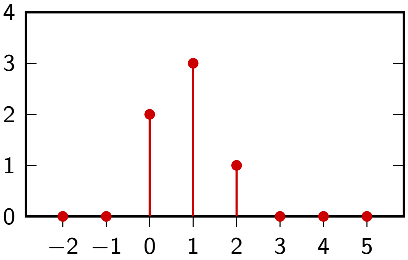
Fundamental result: impulse response fully characterizes the LTI system!

Example

$h[n]$



$x[n]$



$$h[n] = \alpha^n \, u[n]$$

$$x[n] = \begin{cases} 2 & n = 0 \\ 3 & n = 1 \\ 1 & n = 2 \\ 0 & \text{otherwise} \end{cases}$$

Example

- $x[n] = 2\delta[n] + 3\delta[n-1] + \delta[n-2]$
- we know the impulse response $h[n] = \mathcal{H}\{\delta[n]\}$;
- compute $y[n] = \mathcal{H}\{x[n]\}$ exploiting linearity and time-invariance

Example

- $x[n] = 2\delta[n] + 3\delta[n-1] + \delta[n-2]$
- we know the impulse response $h[n] = \mathcal{H}\{\delta[n]\}$;
- compute $y[n] = \mathcal{H}\{x[n]\}$ exploiting linearity and time-invariance

Example

- $x[n] = 2\delta[n] + 3\delta[n-1] + \delta[n-2]$
- we know the impulse response $h[n] = \mathcal{H}\{\delta[n]\}$;
- compute $y[n] = \mathcal{H}\{x[n]\}$ exploiting linearity and time-invariance

Example

$$y[n] = \mathcal{H}\{2\delta[n] + 3\delta[n-1] + \delta[n-2]\}$$

$$= 2\mathcal{H}\{\delta[n]\} + 3\mathcal{H}\{\delta[n-1]\} + \mathcal{H}\{\delta[n-2]\}$$

$$= 2h[n] + 3h[n-1] + h[n-2]$$

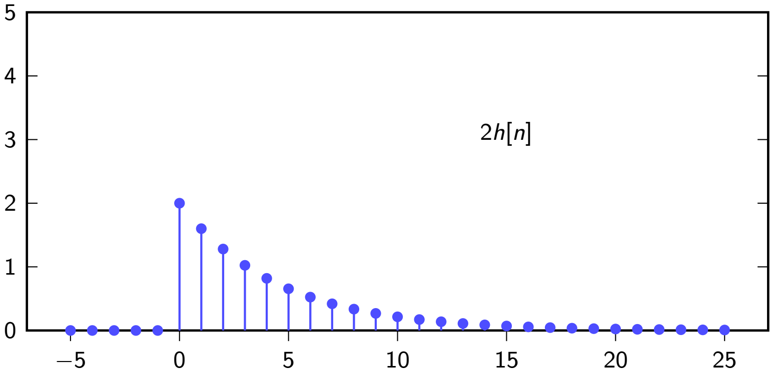
Example

$$
\begin{aligned}
y[n] &= \mathcal{H}\{2\delta[n] + 3\delta[n-1] + \delta[n-2]\} \\
&= 2\mathcal{H}\{\delta[n]\} + 3\mathcal{H}\{\delta[n-1]\} + \mathcal{H}\{\delta[n-2]\} \\
&= 2h[n] + 3h[n-1] + h[n-2]
\end{aligned}
$$

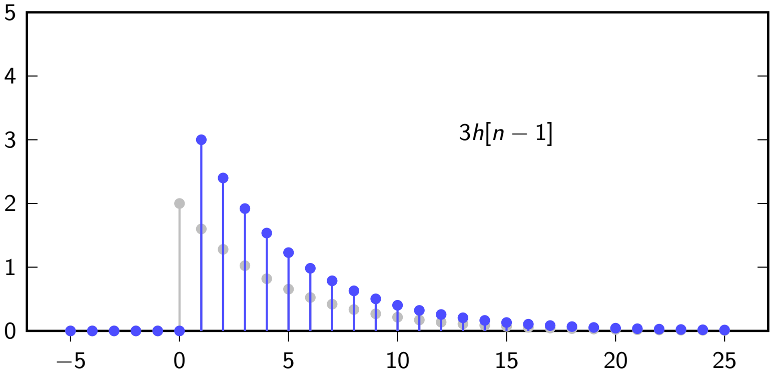
Example

$$
\begin{aligned}
y[n] &= \mathcal{H}\{2\delta[n] + 3\delta[n-1] + \delta[n-2]\} \\
&= 2\mathcal{H}\{\delta[n]\} + 3\mathcal{H}\{\delta[n-1]\} + \mathcal{H}\{\delta[n-2]\} \\
&= 2h[n] + 3h[n-1] + h[n-2]
\end{aligned}
$$

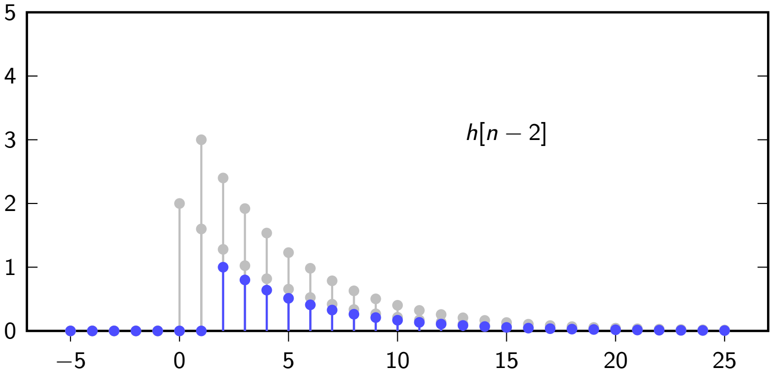
Example



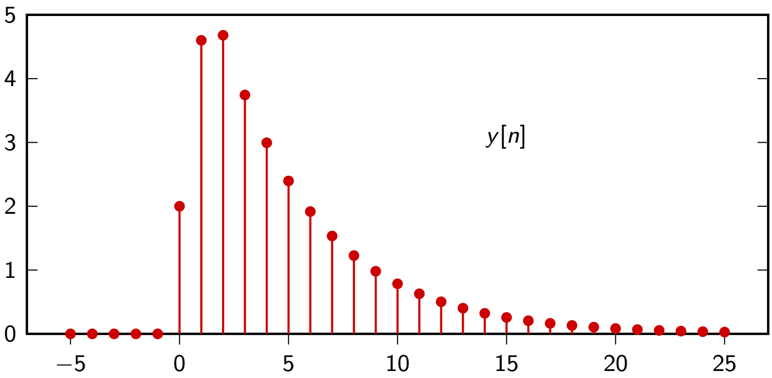
Example



Example



Example



Convolution

We can always write a canonical-base decomposition:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

by linearity and time invariance:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

$$= x[n] * h[n]$$

Convolution

We can always write a canonical-base decomposition:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

by linearity and time invariance:

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} x[k]h[n-k] \\ &= x[n] * h[n] \end{aligned}$$

Performing the convolution algorithmically

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Ingredients:

- a sequence $x[m]$
- a second sequence $h[m]$

The recipe:

- time-reverse $h[m]$
- at each step n (from $-\infty$ to ∞):
  - center the time-reversed $h[m]$ in n (i.e. shift by $-n$)
  - compute the inner product

Performing the convolution algorithmically

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

The recipe:

- time-reverse $h[m]$
- at each step n (from $-\infty$ to ∞):
  - center the time-reversed $h[m]$ in n (i.e. shift by $-n$)
  - compute the inner product

Ingredients:

- a sequence $x[m]$
- a second sequence $h[m]$

Performing the convolution algorithmically

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

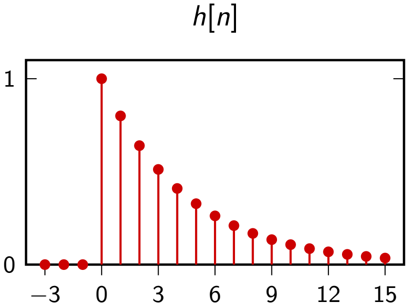
Ingredients:

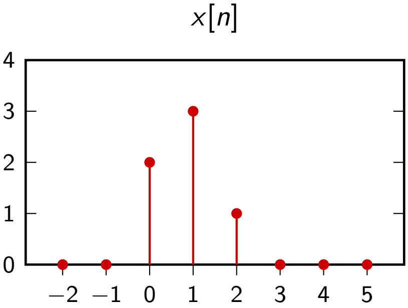
- a sequence $x[m]$
- a second sequence $h[m]$

The recipe:

- time-reverse $h[m]$
- at each step n (from $-\infty$ to ∞):
  - center the time-reversed $h[m]$ in n (i.e. shift by $-n$)
  - compute the inner product

Same example, different perspective

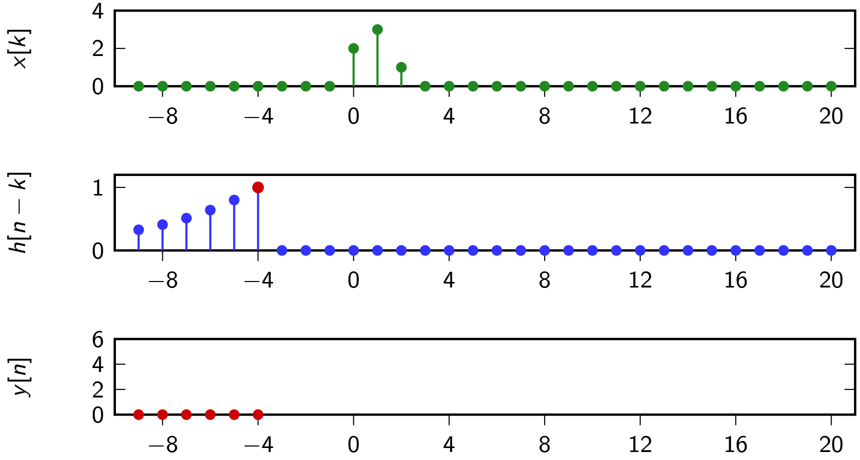




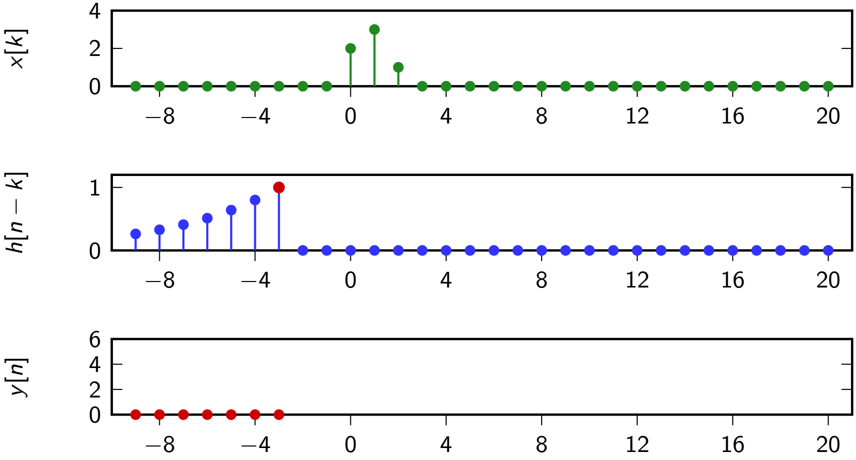
$$h[n] = \alpha^n \, u[n]$$

$$x[n] = \begin{cases} 2 & n = 0 \\ 3 & n = 1 \\ 1 & n = 2 \\ 0 & \text{otherwise} \end{cases}$$

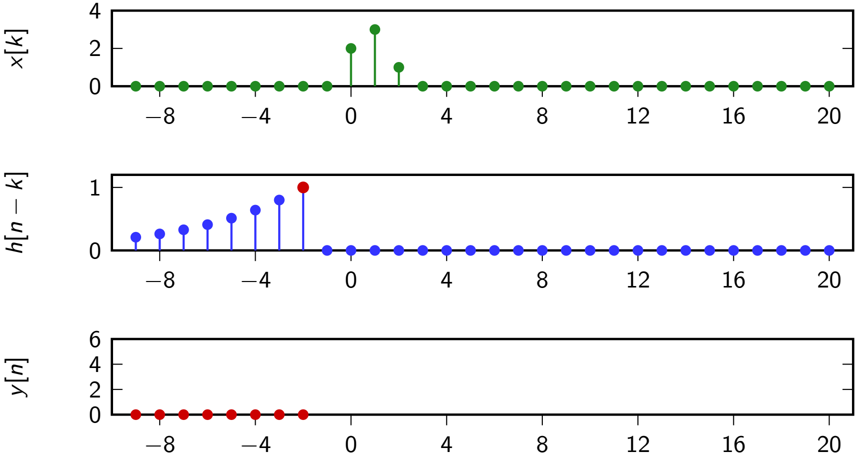
Convolution example



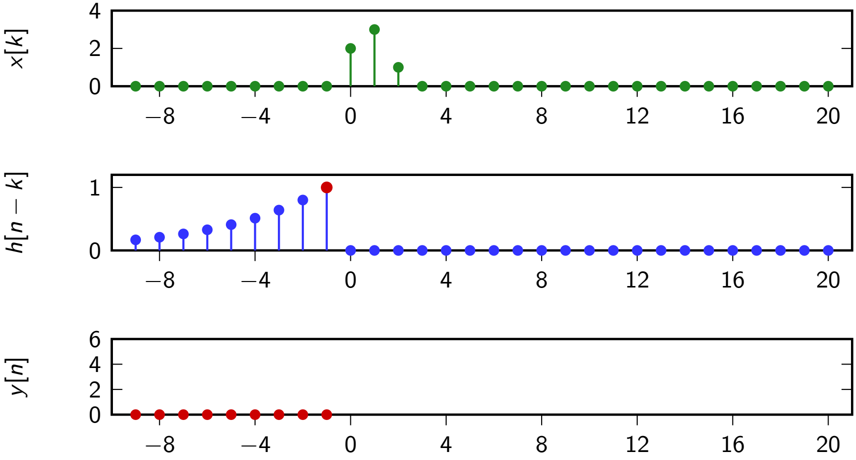
Convolution example



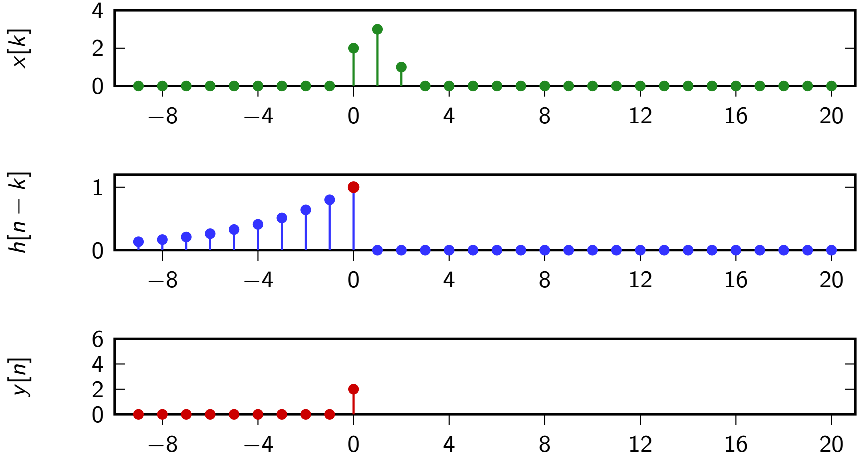
Convolution example



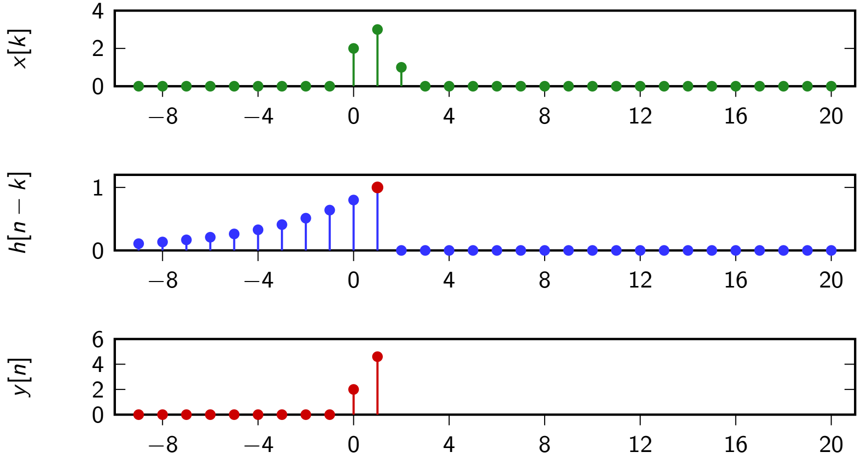
Convolution example



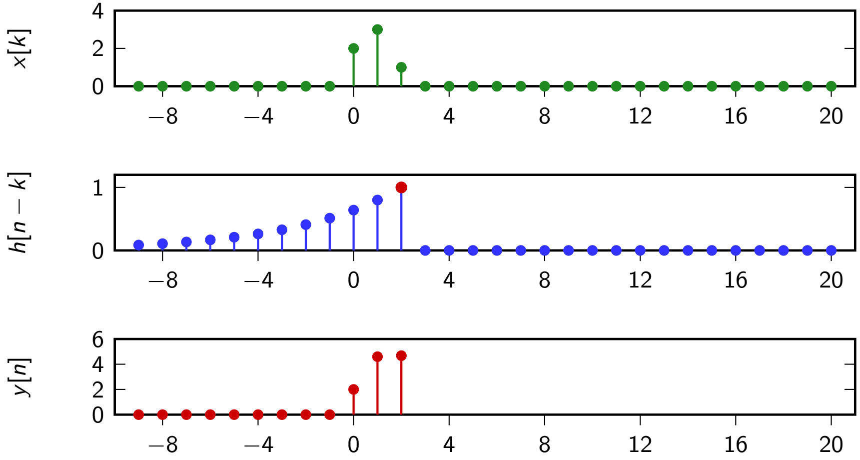
Convolution example



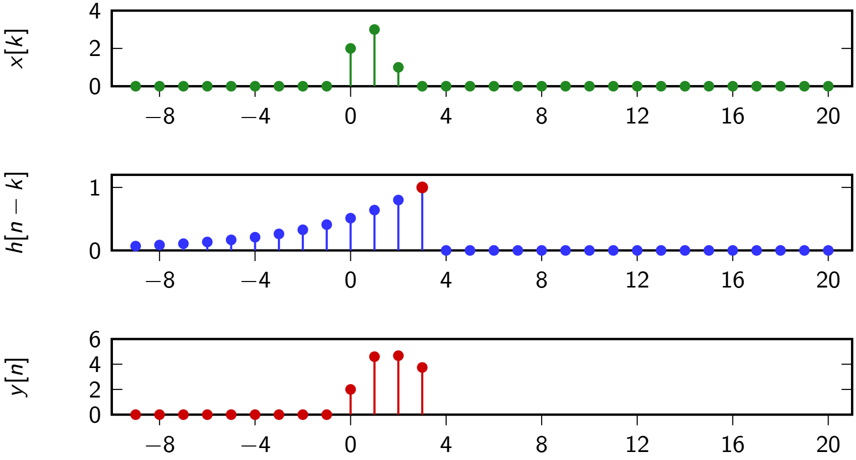
Convolution example



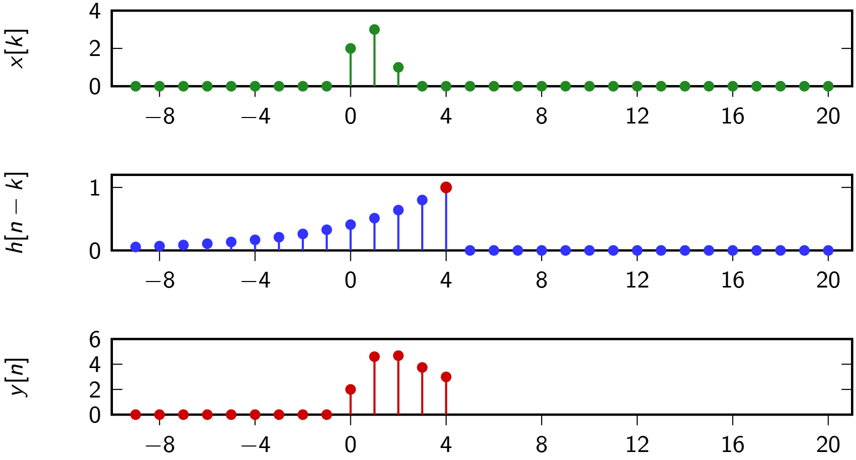
Convolution example



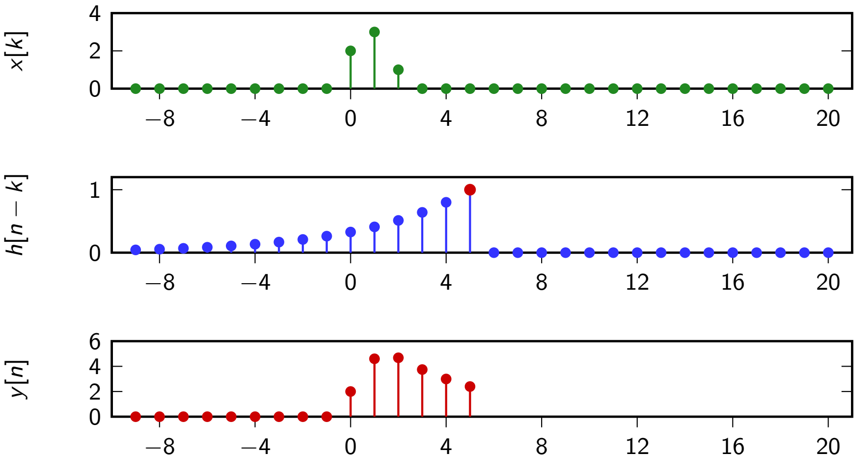
Convolution example



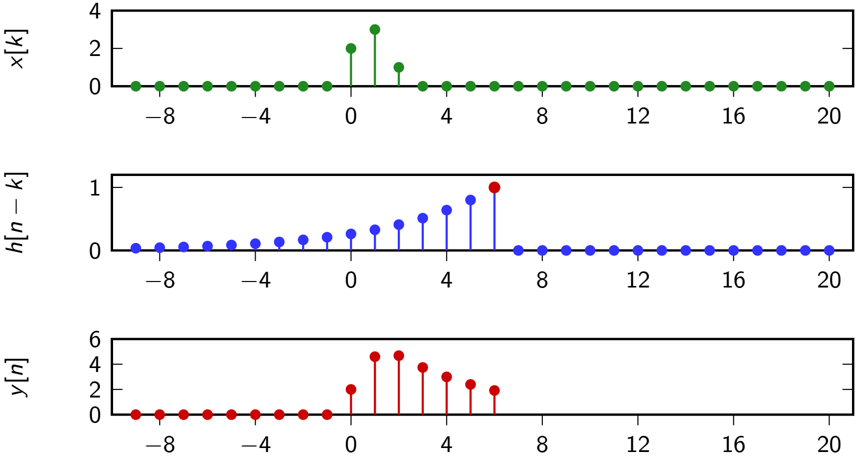
Convolution example



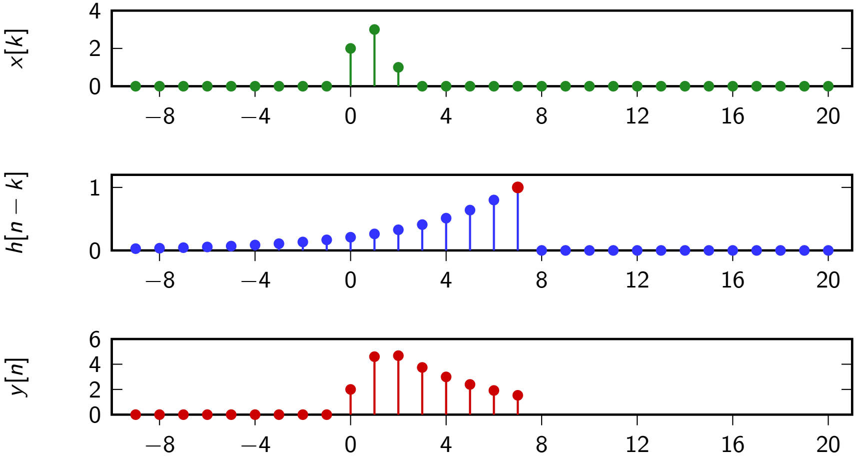
Convolution example



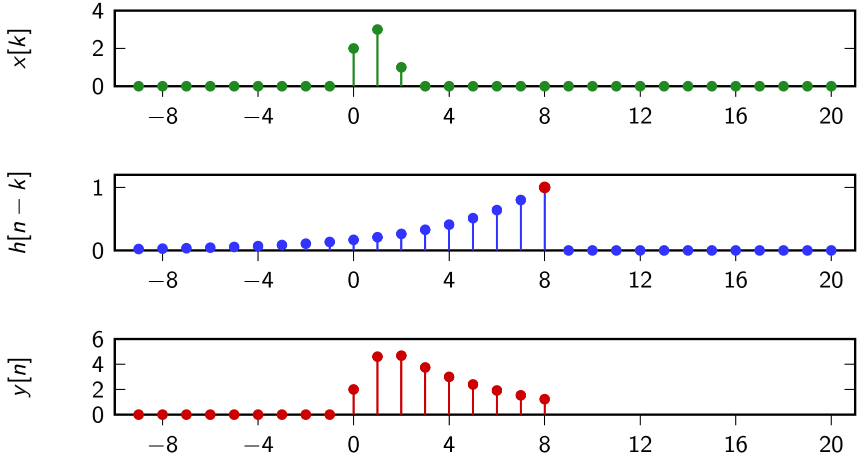
Convolution example



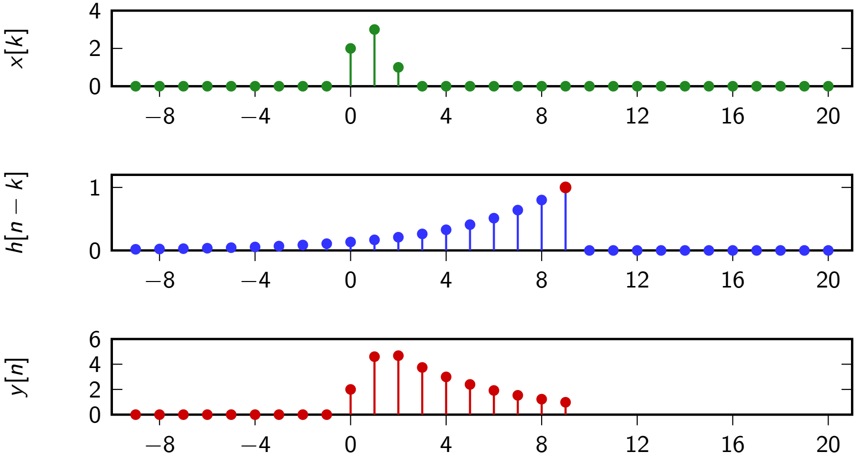
Convolution example



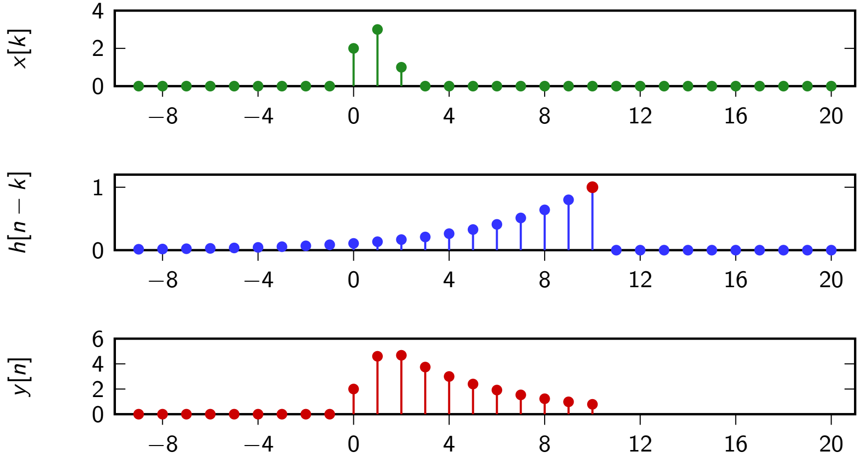
Convolution example



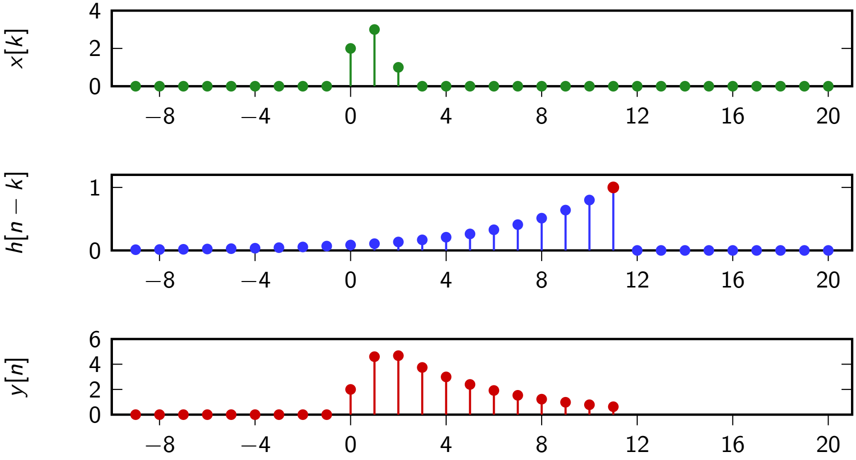
Convolution example



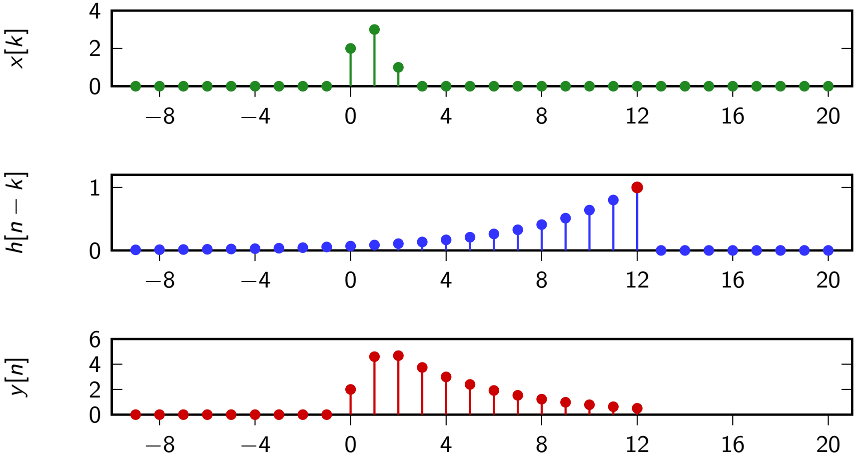
Convolution example



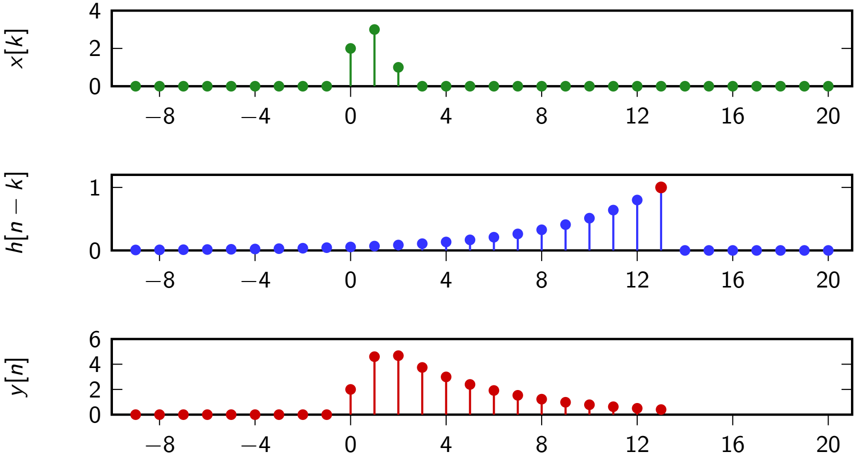
Convolution example



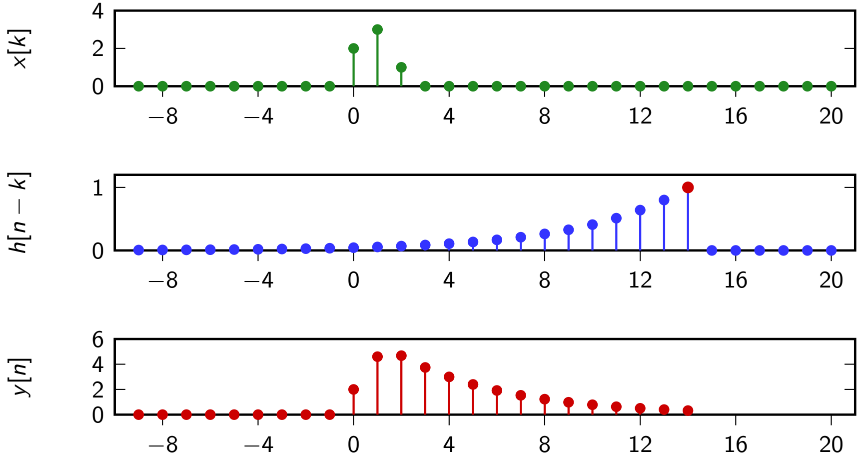
Convolution example



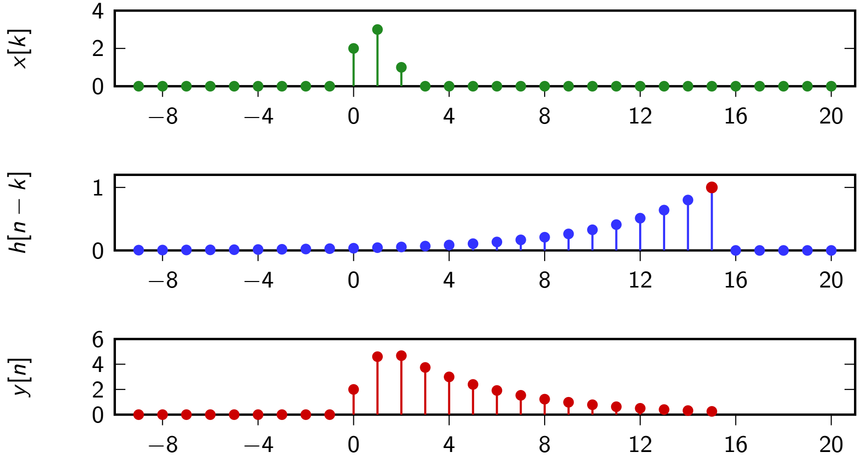
Convolution example



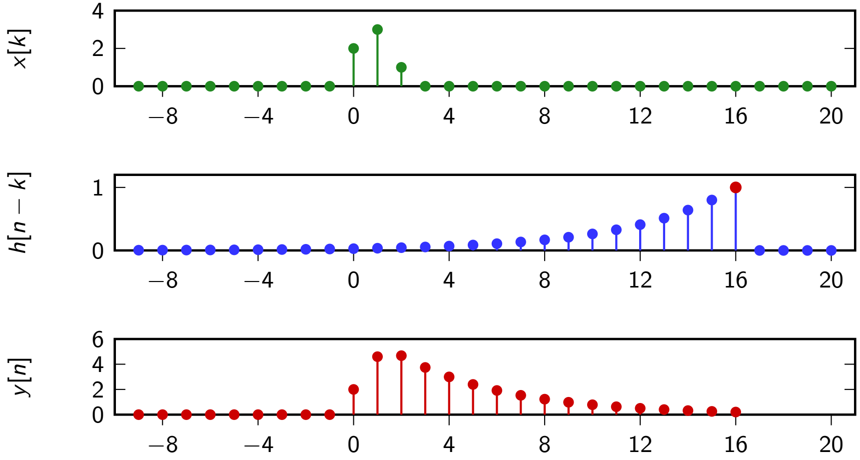
Convolution example



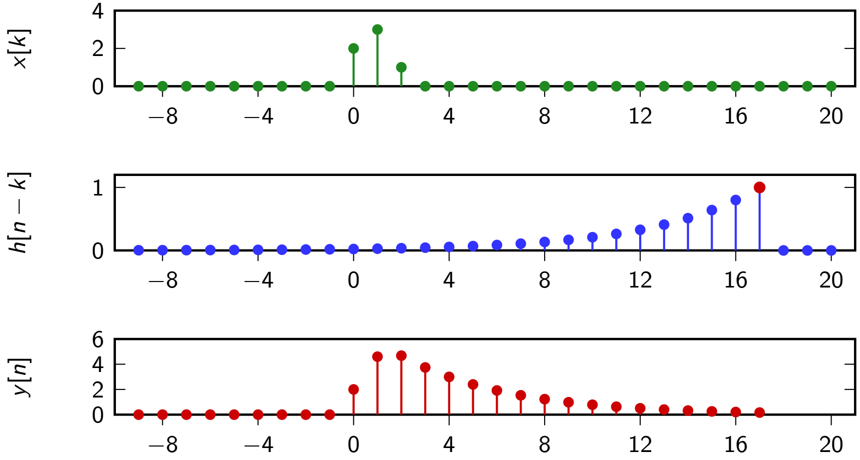
Convolution example



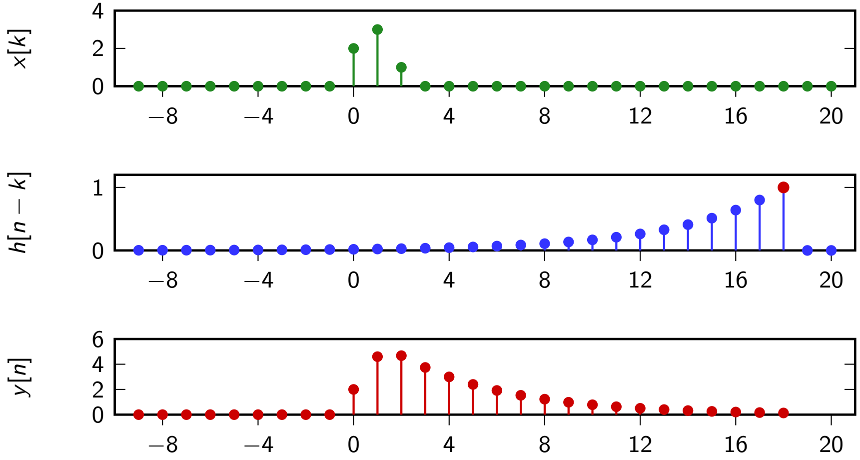
Convolution example



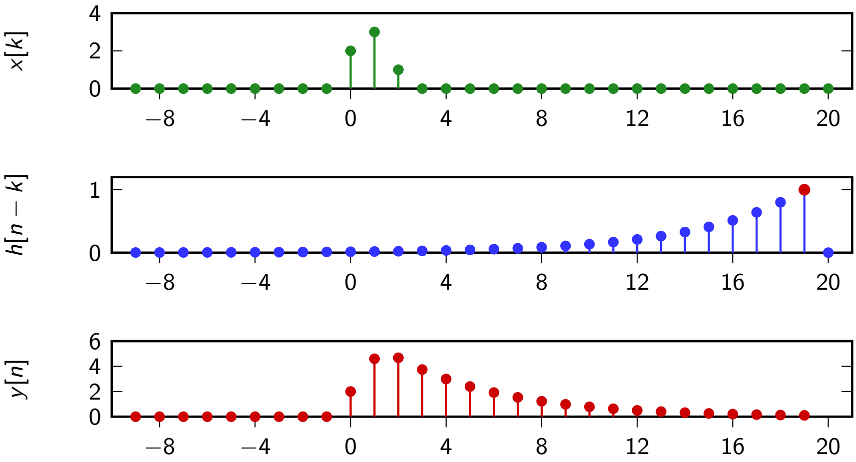
Convolution example



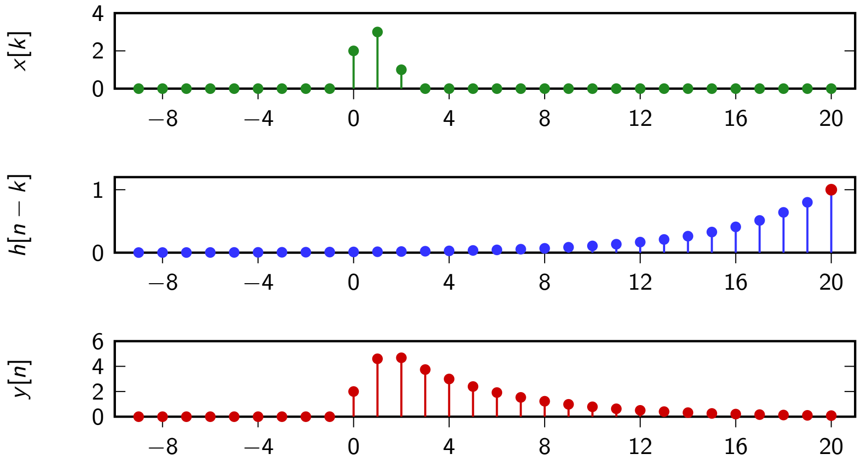
Convolution example



Convolution example



Convolution example



Convolution properties

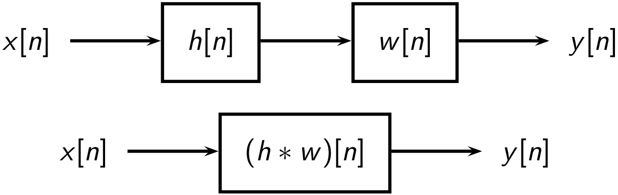
- linearity and time invariance (by definition)
- commutativity: $(x * h)[n] = (h * x)[n]$
- associativity for absolutely- and square-summable sequences: $((x * h) * w)[n] = (x * (h * w))[n]$

Convolution properties

- linearity and time invariance (by definition)
- commutativity: $(x * h)[n] = (h * x)[n]$
- associativity for absolutely- and square-summable sequences: $((x * h) * w)[n] = (x * (h * w))[n]$

Convolution properties

- linearity and time invariance (by definition)
- commutativity: $(x * h)[n] = (h * x)[n]$
- associativity for absolutely- and square-summable sequences:
  $((x * h) * w)[n] = (x * (h * w))[n]$



filtering by example

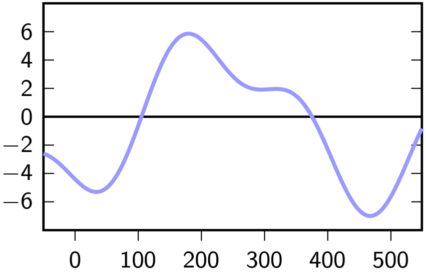
Overview:

- Moving average filter
- Leaky integrator

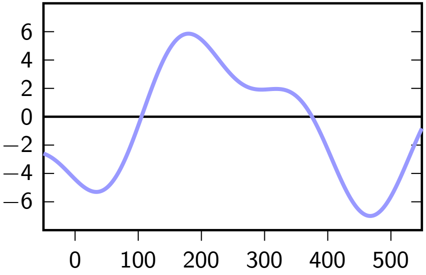
Overview:

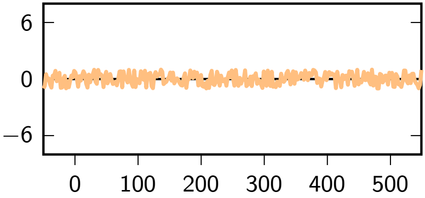
- Moving average filter
- Leaky integrator

Typical filtering scenario: denoising

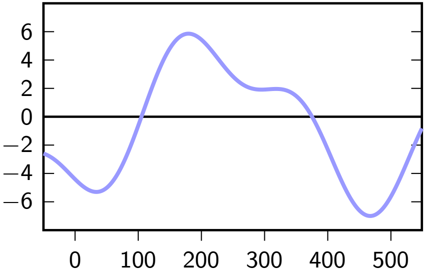


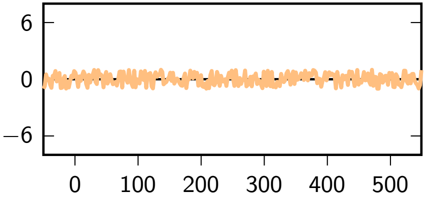
Typical filtering scenario: denoising

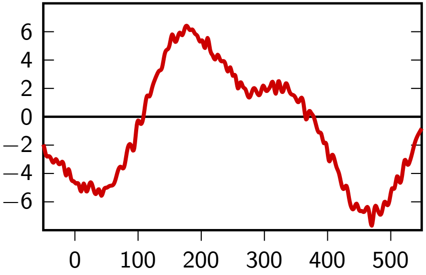




Typical filtering scenario: denoising







Denoising by Moving Average

- idea: replace each sample by the local average
- for instance: $y[n] = (x[n] + x[n-1])/2$
- more general:

$$y[n] = \frac{1}{M}\sum_{k=0}^{M-1} x[n-k]$$

Denoising by Moving Average

- idea: replace each sample by the local average
- for instance: $y[n] = (x[n] + x[n-1])/2)$
- more generally:

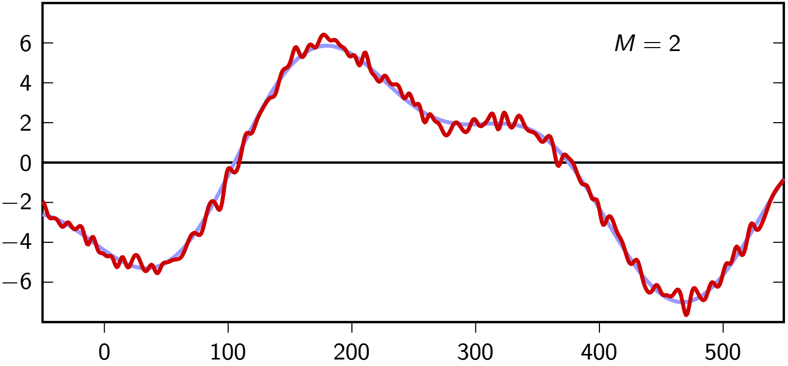
$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k]$$

Denoising by Moving Average

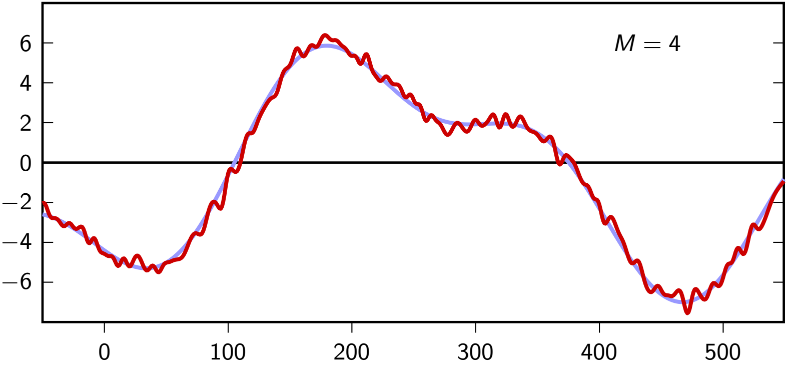
- idea: replace each sample by the local average
- for instance: $y[n] = (x[n] + x[n-1])/2)$
- more generally:

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k]$$

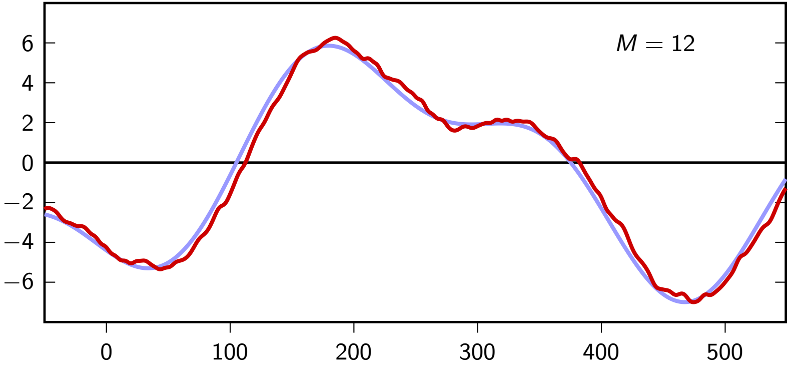
Denoising by Moving Average



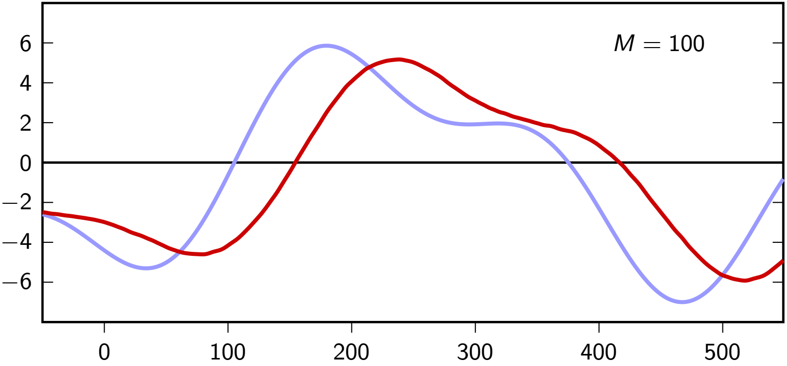
Denoising by Moving Average



Denoising by Moving Average



Denoising by Moving Average



MA: impulse response

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k]$$

MA: impulse response

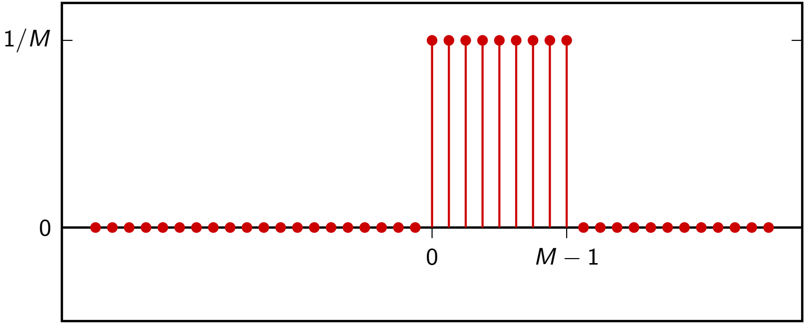
$$h[n] = \frac{1}{M} \sum_{k=0}^{M-1} \delta[n-k]$$

MA: impulse response

$$h[n] = \frac{1}{M} \sum_{k=0}^{M-1} \delta[n-k]$$

$$= \begin{cases} 1/M & \text{for } 0 \leq n < M \\ 0 & \text{otherwise} \end{cases}$$

MA: impulse response



MA: analysis

- smoothing effect proportional to M
- number of operations and storage also proportional to M

From the MA to a first-order recursion

$$y_M[n] = \boxed{\frac{1}{M}\left(x[n] + x[n-1] + x[n-2] + \ldots + x[n-M+1]\right)}$$

moving average over M points

From the MA to a first-order recursion

$$y_M[n] = \boxed{\frac{1}{M}\left(x[n] + x[n-1] + x[n-2] + \ldots + x[n-M+1]\right)}$$

moving average over M points

$$y_{M-1}[n] = \frac{1}{M-1}\left(x[n] + x[n-1] + x[n-2] + \ldots + x[n-M+2]\right)$$

From the MA to a first-order recursion

$$y_M[n] = \frac{1}{M}x[n] + \frac{1}{M}\left(x[n-1] + x[n-2] + \ldots + x[n-M+1]\right)$$

From the MA to a first-order recursion

$$y_M[n] = \frac{1}{M}x[n] + \boxed{\frac{1}{M}\left(x[n-1] + x[n-2] + \ldots + x[n-M+1]\right)}$$

"almost" $y_{M-1}[n-1]$

i.e., moving average over $M-1$ points, delayed by one

From the MA to a first-order recursion

$$y_M[n] = \frac{1}{M}x[n] + \boxed{\frac{1}{M}\left(x[n-1] + x[n-2] + \ldots + x[n-M+1]\right)}$$

$$y_M[n] = \frac{1}{M}x[n] + \frac{M-1}{M}y_{M-1}[n-1]$$

From the MA to a first-order recursion

$$y_M[n] = \frac{M-1}{M} y_{M-1}[n-1] + \frac{1}{M} x[n]$$

$$y_M[n] = \lambda y_{M-1}[n-1] + (1-\lambda) x[n], \qquad \lambda = \frac{M-1}{M}$$

From the MA to a first-order recursion

$$y_M[n] = \frac{M-1}{M} y_{M-1}[n-1] + \frac{1}{M} x[n]$$

$$y_M[n] = \lambda y_{M-1}[n-1] + (1-\lambda)x[n], \qquad \lambda = \frac{M-1}{M}$$

The Leaky Integrator

- when M is large, $y_{M-1}[n] \approx y_M[n]$ (and $\lambda \approx 1$)
- try the filter

$$y[n] = \lambda y[n-1] + (1-\lambda)x[n]$$

- filter is now recursive, since it uses its previous output value

The Leaky Integrator

- when M is large, $y_{M-1}[n] \approx y_M[n]$ (and $\lambda \approx 1$)
- try the filter

$$y[n] = \lambda y[n-1] + (1-\lambda)x[n]$$

- filter is now recursive, since it uses its previous output value

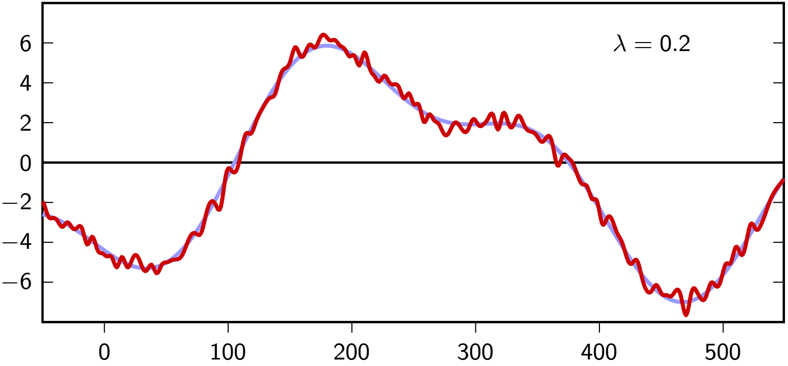
The Leaky Integrator

- when M is large, $y_{M-1}[n] \approx y_M[n]$ (and $\lambda \approx 1$)
- try the filter

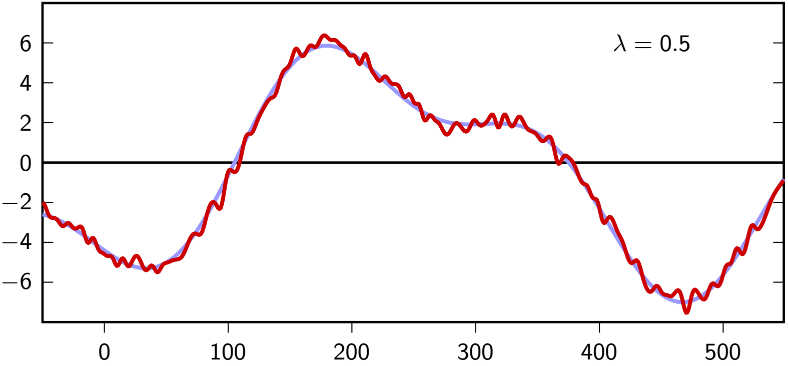
$$y[n] = \lambda y[n-1] + (1-\lambda)x[n]$$

- filter is now recursive, since it uses its previous output value

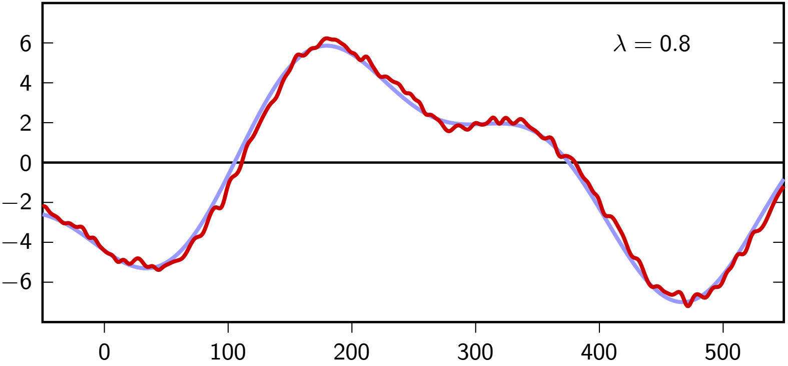
Denoising recursively with the Leaky Integrator



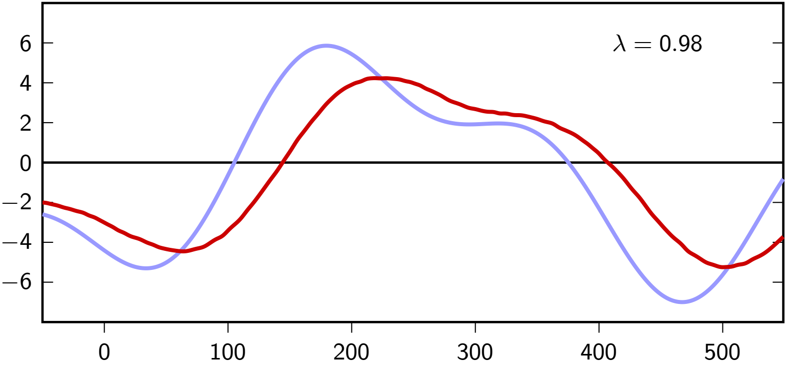
Denoising recursively with the Leaky Integrator



Denoising recursively with the Leaky Integrator



Denoising recursively with the Leaky Integrator



What about the impulse response?

$$y[n] = \lambda y[n-1] + (1-\lambda)\delta[n]$$

- $y[n] = 0$ for all $n < 0$
- $y[0] = \lambda y[-1] + (1-\lambda)\delta[0] = (1-\lambda)$
- $y[1] = \lambda y[0] + (1-\lambda)\delta[1] = \lambda(1-\lambda)$
- $y[2] = \lambda y[1] + (1-\lambda)\delta[2] = \lambda^2(1-\lambda)$
- $y[3] = \lambda y[2] + (1-\lambda)\delta[3] = \lambda^3(1-\lambda)$
- $\ldots$

What about the impulse response?

$$y[n] = \lambda y[n-1] + (1-\lambda)\delta[n]$$

- $y[n] = 0$ for all $n < 0$
- $y[0] = \lambda y[-1] + (1-\lambda)\delta[0] = (1-\lambda)$
- $y[1] = \lambda y[0] + (1-\lambda)\delta[1] = \lambda(1-\lambda)$
- $y[2] = \lambda y[1] + (1-\lambda)\delta[2] = \lambda^2(1-\lambda)$
- $y[3] = \lambda y[2] + (1-\lambda)\delta[3] = \lambda^3(1-\lambda)$
- $\ldots$

What about the impulse response?

$$y[n] = \lambda y[n-1] + (1-\lambda)\delta[n]$$

- $y[n] = 0$ for all $n < 0$
- $y[0] = \lambda y[-1] + (1-\lambda)\delta[0] = (1-\lambda)$
- $y[1] = \lambda y[0] + (1-\lambda)\delta[1] = \lambda(1-\lambda)$
- $y[2] = \lambda y[1] + (1-\lambda)\delta[2] = \lambda^2(1-\lambda)$
- $y[3] = \lambda y[2] + (1-\lambda)\delta[3] = \lambda^3(1-\lambda)$
- $\ldots$

What about the impulse response?

$$y[n] = \lambda y[n-1] + (1-\lambda)\delta[n]$$

- $y[n] = 0$ for all $n < 0$
- $y[0] = \lambda y[-1] + (1-\lambda)\delta[0] = (1-\lambda)$
- $y[1] = \lambda y[0] + (1-\lambda)\delta[1] = \lambda(1-\lambda)$
- $y[2] = \lambda y[1] + (1-\lambda)\delta[2] = \lambda^2(1-\lambda)$
- $y[3] = \lambda y[2] + (1-\lambda)\delta[3] = \lambda^3(1-\lambda)$
- $\ldots$

What about the impulse response?

$$y[n] = \lambda y[n-1] + (1-\lambda)\delta[n]$$

- $y[n] = 0$ for all $n < 0$
- $y[0] = \lambda y[-1] + (1-\lambda)\delta[0] = (1-\lambda)$
- $y[1] = \lambda y[0] + (1-\lambda)\delta[1] = \lambda(1-\lambda)$
- $y[2] = \lambda y[1] + (1-\lambda)\delta[2] = \lambda^2(1-\lambda)$
- $y[3] = \lambda y[2] + (1-\lambda)\delta[3] = \lambda^3(1-\lambda)$
- ...

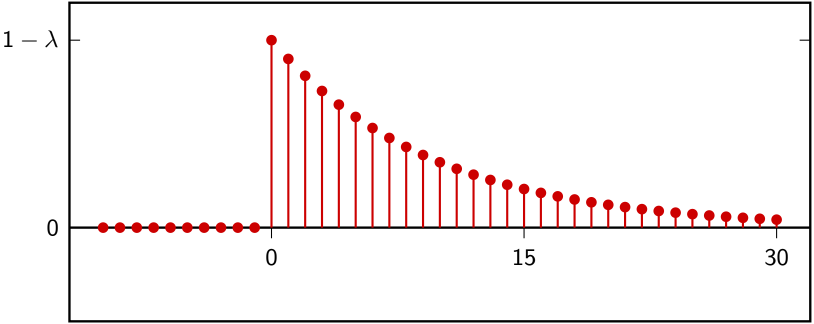
What about the impulse response?

$$y[n] = \lambda y[n-1] + (1-\lambda)\delta[n]$$

- $y[n] = 0$ for all $n < 0$
- $y[0] = \lambda y[-1] + (1-\lambda)\delta[0] = (1-\lambda)$
- $y[1] = \lambda y[0] + (1-\lambda)\delta[1] = \lambda(1-\lambda)$
- $y[2] = \lambda y[1] + (1-\lambda)\delta[2] = \lambda^2(1-\lambda)$
- $y[3] = \lambda y[2] + (1-\lambda)\delta[3] = \lambda^3(1-\lambda)$
- $\ldots$

Impulse response

$$h[n] = (1-\lambda)\lambda^n\, u[n]$$



Leaky Integrator: why the name

Discrete-time integrator is a boundless accumulator:

$$y[n] = \sum_{k=-\infty}^{n} x[k]$$

We can rewrite the integrator as

$$y[n] = y[n-1] + x[n]$$

Leaky Integrator: why the name

To prevent "explosion" pick $\lambda < 1$

$$y[n] = \lambda y[n-1] + (1-\lambda)x[n]$$

keep only a fraction λ of the accumulated value so far and forget ("leak") a fraction $1-\lambda$

add only a fraction $1-\lambda$ of the current value to the accumulator

LI: analysis

- smoothing effect dependent on λ
- number of operations and storage: *independent of* λ
- recursion generates infinite-length impulse response
- infinite-length impulse responses are computable

filter stability

Filter types according to impulse response

- Finite Impulse Response (FIR)
- Infinite Impulse Response (IIR)
- causal
- noncausal

Filter types according to impulse response

- Finite Impulse Response (FIR)
- Infinite Impulse Response (IIR)
- causal
- noncausal

Filter types according to impulse response

- Finite Impulse Response (FIR)
- Infinite Impulse Response (IIR)
- causal
- noncausal

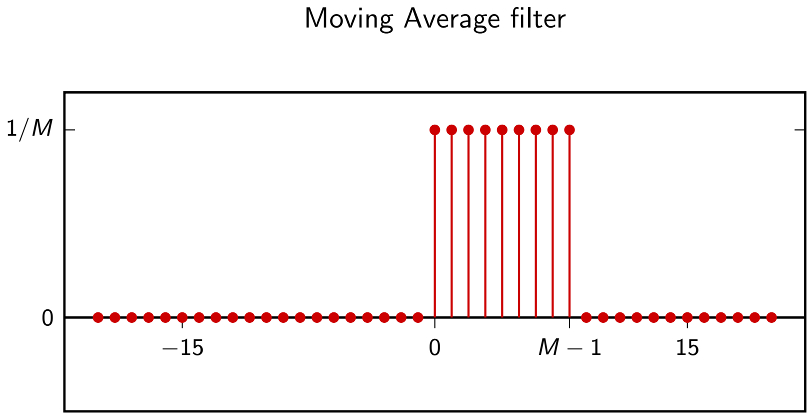
Filter types according to impulse response

- Finite Impulse Response (FIR)
- Infinite Impulse Response (IIR)
- causal
- noncausal

FIR

- impulse response has finite support
- only a finite number of samples are involved in the computation of each output sample

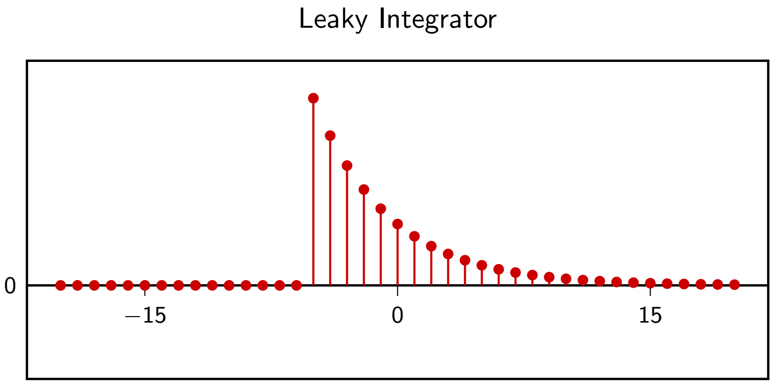
FIR (example)



IIR

- impulse response has infinite support
- a potentially infinite number of samples are involved in the computation of each output sample
- surprisingly, in many cases the computation can still be performed in a finite amount of steps

IIR (example)

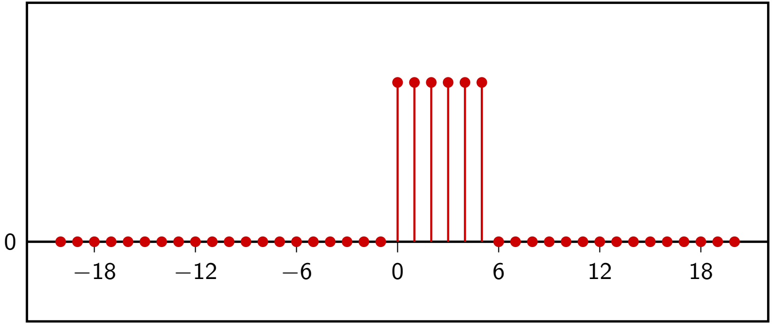


Causal vs Noncausal

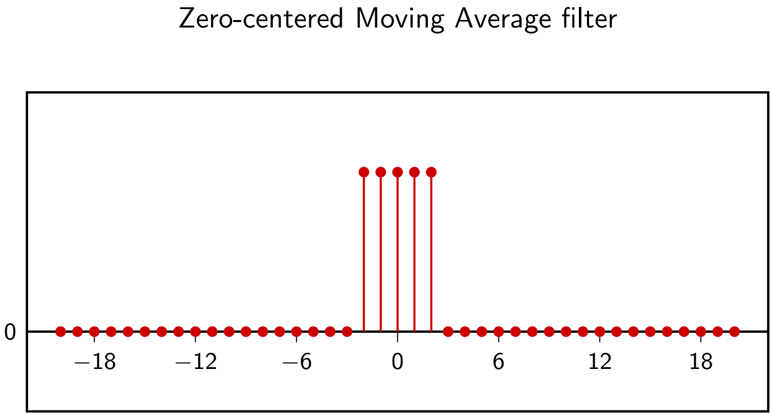
- causal:
  - impulse response is zero for $n < 0$
  - only past samples (with respect to the present) are involved in the computation of each output sample
  - causal filters can work "on line" since they only need the past
- noncausal:
  - impulse response is nonzero for some (or all) $n < 0$
  - can still be implemented in a offline fashion (when all input data is available on storage, e.g. in Image Processing)

Causal example

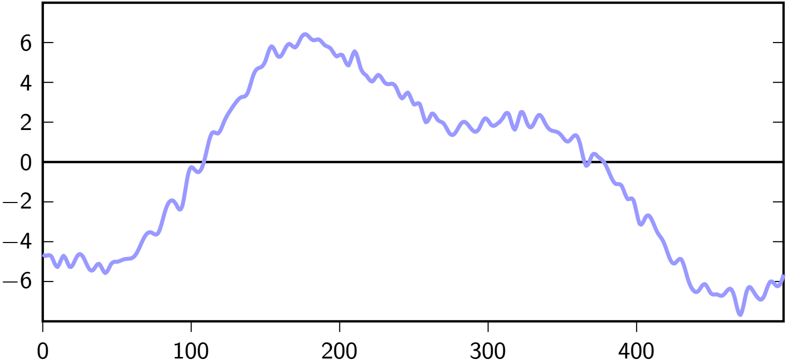
Moving Average filter



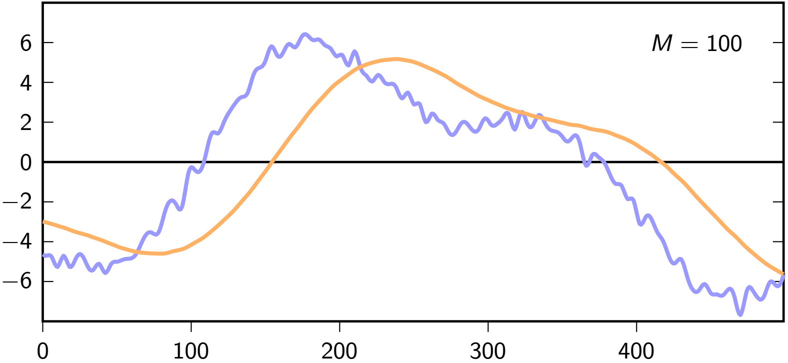
Noncausal example



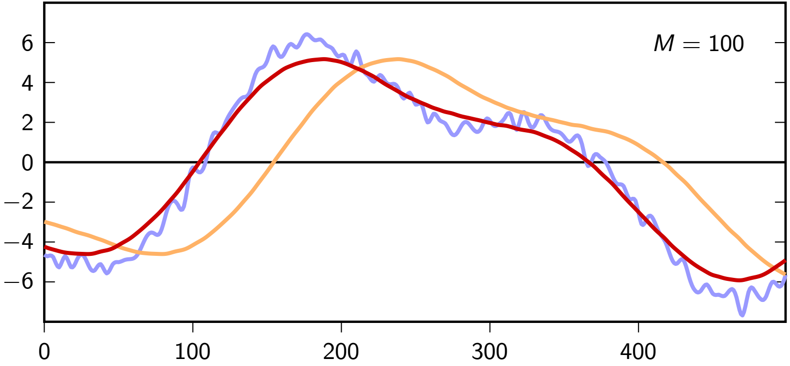
Causal and Noncausal Moving Average



Causal and Noncausal Moving Average



Causal and Noncausal Moving Average



Stability

- key concept: avoid "explosions" if the input is nice
- a nice signal is a bounded signal: $|x[n]| < M$ for all n
- Bounded-Input Bounded-Output (BIBO) stability: if the input is nice the output should be nice

Stability

- key concept: avoid "explosions" if the input is nice
- a nice signal is a bounded signal: $|x[n]| < M$ for all n
- Bounded-Input Bounded-Output (BIBO) stability: if the input is nice the output should be nice

Stability

- key concept: avoid "explosions" if the input is nice
- a nice signal is a bounded signal: $|x[n]| < M$ for all n
- Bounded-Input Bounded-Output (BIBO) stability: if the input is nice the output should be nice

Fundamental Stability Theorem

A filter is BIBO stable if and only if its impulse response is absolutely summable

Proof ($\Rightarrow$)

Hypotheses:

- $|x[n]| < M$
- $\sum_n |h[n]| = L < \infty$

Thesis:

- $|y[n]| < \infty$

Proof:

$$\begin{aligned} |y[n]| &= \left| \sum_{k=-\infty}^{\infty} h[k]x[n-k] \right| \\ &\leq \sum_{k=-\infty}^{\infty} |h[k]x[n-k]| \\ &\leq M \sum_{k=-\infty}^{\infty} |h[k]| \\ &\leq ML \end{aligned}$$

Proof ($\Rightarrow$)

Hypotheses:

- $|x[n]| < M$
- $\sum_n |h[n]| = L < \infty$

Thesis:

- $|y[n]| < \infty$

Proof:

$$|y[n]| = \left| \sum_{k=-\infty}^{\infty} h[k]x[n-k] \right|$$

$$\leq \sum_{k=-\infty}^{\infty} |h[k]x[n-k]|$$

$$\leq M \sum_{k=-\infty}^{\infty} |h[k]|$$

$$\leq ML$$

Proof ($\Rightarrow$)

Hypotheses:

- $|x[n]| < M$
- $\sum_n |h[n]| = L < \infty$

Thesis:

- $|y[n]| < \infty$

Proof:

$$|y[n]| = \left| \sum_{k=-\infty}^{\infty} h[k]x[n-k] \right|$$

$$\leq \sum_{k=-\infty}^{\infty} |h[k]x[n-k]|$$

$$\leq M \sum_{k=-\infty}^{\infty} |h[k]|$$

$$\leq ML$$

Proof ($\Rightarrow$)

Hypotheses:

- $|x[n]| < M$
- $\sum_n |h[n]| = L < \infty$

Thesis:

- $|y[n]| < \infty$

Proof:

$$|y[n]| = \left| \sum_{k=-\infty}^{\infty} h[k]x[n-k] \right|$$

$$\leq \sum_{k=-\infty}^{\infty} |h[k]x[n-k]|$$

$$\leq M \sum_{k=-\infty}^{\infty} |h[k]|$$

$$\leq ML$$

Proof ($\Rightarrow$)

Hypotheses:

- $|x[n]| < M$
- $\sum_n |h[n]| = L < \infty$

Thesis:

- $|y[n]| < \infty$

Proof:

$$
\begin{aligned}
|y[n]| &= \left| \sum_{k=-\infty}^{\infty} h[k]x[n-k] \right| \\
&\leq \sum_{k=-\infty}^{\infty} |h[k]x[n-k]| \\
&\leq M \sum_{k=-\infty}^{\infty} |h[k]| \\
&\leq ML
\end{aligned}
$$

Proof ($\Leftarrow$)

Hypothesis:

- $\forall |x[n]| < \infty$, $|(x * h)[n]| < \infty$

Thesis:

- $\sum_n |h[n]| < \infty$

Proof (by contradiction):

- assume hypothesis true, yet $\sum_n |h[n]| = \infty$
- build $x[n] = \begin{cases} +1 & \text{if } h[-n] \geq 0 \\ -1 & \text{if } h[-n] < 0 \end{cases}$
- clearly, $|x[n]| < \infty$
- however

$$(x * h)[0] = \sum_{k=-\infty}^{\infty} h[k]x[-k] = \sum_{k=-\infty}^{\infty} |h[k]| = \infty$$

Proof ($\Leftarrow$)

Hypothesis:

- $\forall |x[n]| < \infty$,
  $|(x * h)[n]| < \infty$

Thesis:

- $\sum_n |h[n]| < \infty$

Proof (by contradiction):

- assume hypothesis true, yet $\sum_n |h[n]| = \infty$
- build $x[n] = \begin{cases} +1 & \text{if } h[-n] \geq 0 \\ -1 & \text{if } h[-n] < 0 \end{cases}$
- clearly, $|x[n]| < \infty$
- however

$$(x * h)[0] = \sum_{k=-\infty}^{\infty} h[k]x[-k] = \sum_{k=-\infty}^{\infty} |h[k]| = \infty$$

Proof ($\Leftarrow$)

Hypothesis:

- $\forall |x[n]| < \infty$, $|(x * h)[n]| < \infty$

Thesis:

- $\sum_n |h[n]| < \infty$

Proof (by contradiction):

- assume hypothesis true, yet $\sum_n |h[n]| = \infty$
- build $x[n] = \begin{cases} +1 & \text{if } h[-n] \geq 0 \\ -1 & \text{if } h[-n] < 0 \end{cases}$
- clearly, $|x[n]| < \infty$
- however

$$(x * h)[0] = \sum_{k=-\infty}^{\infty} h[k]x[-k] = \sum_{k=-\infty}^{\infty} |h[k]| = \infty$$

Proof ($\Leftarrow$)

Hypothesis:

- $\forall |x[n]| < \infty$,
  $|(x * h)[n]| < \infty$

Thesis:

- $\sum_n |h[n]| < \infty$

Proof (by contradiction):

- assume hypothesis true, yet $\sum_n |h[n]| = \infty$
- build $x[n] = \begin{cases} +1 & \text{if } h[-n] \geq 0 \\ -1 & \text{if } h[-n] < 0 \end{cases}$
- clearly, $|x[n]| < \infty$
- however

$$(x * h)[0] = \sum_{k=-\infty}^{\infty} h[k]x[-k] = \sum_{k=-\infty}^{\infty} |h[k]| = \infty$$

Proof ($\Leftarrow$)

Hypothesis:

- $\forall |x[n]| < \infty$,
  $|(x * h)[n]| < \infty$

Thesis:

- $\sum_n |h[n]| < \infty$

Proof (by contradiction):

- assume hypothesis true, yet $\sum_n |h[n]| = \infty$
- build $x[n] = \begin{cases} +1 & \text{if } h[-n] \geq 0 \\ -1 & \text{if } h[-n] < 0 \end{cases}$
- clearly, $|x[n]| < \infty$
- however

$$(x * h)[0] = \sum_{k=-\infty}^{\infty} h[k]x[-k] = \sum_{k=-\infty}^{\infty} |h[k]| = \infty$$

The good news

FIR filters are always stable

Checking the stability of IIRs

Let's check the Leaky Integrator:

$$\sum_{n=-\infty}^{\infty} |h[n]| = |1-\lambda| \sum_{n=0}^{\infty} |\lambda|^n$$

$$= \lim_{n\to\infty} |1-\lambda| \frac{1-|\lambda|^{n+1}}{1-|\lambda|}$$

$$< \infty \quad \text{for } |\lambda| < 1$$

stability is guaranteed for $|\lambda| < 1$

Checking the stability of IIRs

Let's check the Leaky Integrator:

$$\sum_{n=-\infty}^{\infty} |h[n]| = |1-\lambda| \sum_{n=0}^{\infty} |\lambda|^n$$

$$= \lim_{n\to\infty} |1-\lambda| \frac{1-|\lambda|^{n+1}}{1-|\lambda|}$$

$$< \infty \quad \text{for } |\lambda| < 1$$

stability is guaranteed for $|\lambda| < 1$

Checking the stability of IIRs

Let's check the Leaky Integrator:

$$\sum_{n=-\infty}^{\infty} |h[n]| = |1-\lambda| \sum_{n=0}^{\infty} |\lambda|^n$$

$$= \lim_{n\to\infty} |1-\lambda| \frac{1-|\lambda|^{n+1}}{1-|\lambda|}$$

$$< \infty \quad \text{for } |\lambda| < 1$$

stability is guaranteed for $|\lambda| < 1$

Checking the stability of IIRs

Let's check the Leaky Integrator:

$$\sum_{n=-\infty}^{\infty} |h[n]| = |1-\lambda| \sum_{n=0}^{\infty} |\lambda|^n$$

$$= \lim_{n\to\infty} |1-\lambda| \frac{1-|\lambda|^{n+1}}{1-|\lambda|}$$

$$< \infty \quad \text{for } |\lambda| < 1$$

stability is guaranteed for $|\lambda| < 1$

Checking the stability of IIRs

We will study indirect methods for filter stability later

frequency response

Overview:

- Eigensequences
- Convolution theorem
- Frequency and phase response

Overview:

- Eigensequences
- Convolution theorem
- Frequency and phase response

Overview:

- Eigensequences
- Convolution theorem
- Frequency and phase response

A remarkable result

$$e^{j\omega_0 n} \longrightarrow \boxed{\mathcal{H}} \longrightarrow ?$$

A remarkable result

$$
\begin{aligned}
y[n] &= e^{j\omega_0 n} * h[n] \\
&= h[n] * e^{j\omega_0 n} \\
&= \sum_{k=-\infty}^{\infty} h[k] e^{j\omega_0 (n-k)} \\
&= e^{j\omega_0 n} \sum_{k=-\infty}^{\infty} h[k] e^{-j\omega_0 k} \\
&= H(e^{j\omega_0})\, e^{j\omega_0 n}
\end{aligned}
$$

A remarkable result

$$
\begin{aligned}
y[n] &= e^{j\omega_0 n} * h[n] \\
&= h[n] * e^{j\omega_0 n} \\
&= \sum_{k=-\infty}^{\infty} h[k] e^{j\omega_0 (n-k)} \\
&= e^{j\omega_0 n} \sum_{k=-\infty}^{\infty} h[k] e^{-j\omega_0 k} \\
&= H(e^{j\omega_0})\, e^{j\omega_0 n}
\end{aligned}
$$

A remarkable result

$$
\begin{aligned}
y[n] &= e^{j\omega_0 n} * h[n] \\
&= h[n] * e^{j\omega_0 n} \\
&= \sum_{k=-\infty}^{\infty} h[k] e^{j\omega_0 (n-k)} \\
&= e^{j\omega_0 n} \sum_{k=-\infty}^{\infty} h[k] e^{-j\omega_0 k} \\
&= H(e^{j\omega_0})\, e^{j\omega_0 n}
\end{aligned}
$$

A remarkable result

$$
\begin{aligned}
y[n] &= e^{j\omega_0 n} * h[n] \\
&= h[n] * e^{j\omega_0 n} \\
&= \sum_{k=-\infty}^{\infty} h[k] e^{j\omega_0 (n-k)} \\
&= e^{j\omega_0 n} \sum_{k=-\infty}^{\infty} h[k] e^{-j\omega_0 k} \\
&= H(e^{j\omega_0})\, e^{j\omega_0 n}
\end{aligned}
$$

A remarkable result

$$
\begin{aligned}
y[n] &= e^{j\omega_0 n} * h[n] \\
&= h[n] * e^{j\omega_0 n} \\
&= \sum_{k=-\infty}^{\infty} h[k] e^{j\omega_0 (n-k)} \\
&= e^{j\omega_0 n} \sum_{k=-\infty}^{\infty} h[k] e^{-j\omega_0 k} \\
&= H(e^{j\omega_0})\, e^{j\omega_0 n}
\end{aligned}
$$

A remarkable result

$$e^{j\omega_0 n} \longrightarrow \boxed{\mathcal{H}} \longrightarrow H(e^{j\omega_0})\, e^{j\omega_0 n}$$

- complex exponentials are *eigensequences* of LTI systems, i.e., linear filters cannot change the frequency of sinusoids
- DTFT of impulse response determines the frequency characteristic of a filter

A remarkable result

$$e^{j\omega_0 n} \longrightarrow \boxed{\mathcal{H}} \longrightarrow H(e^{j\omega_0})\, e^{j\omega_0 n}$$

- complex exponentials are *eigensequences* of LTI systems, i.e., linear filters cannot change the frequency of sinusoids
- DTFT of impulse response determines the frequency characteristic of a filter

Magnitude and phase

If $H(e^{j\omega_0}) = Ae^{j\theta}$, then

$$\mathcal{H}\{e^{j\omega_0 n}\} = Ae^{j(\omega_0 n+\theta)}$$

amplitude:
amplification ($A > 1$)
or attenuation ($0 \leq A < 1$)

phase shift:
delay ($\theta < 0$)
or advancement ($\theta > 0$)

The convolution theorem

In general:

$$\mathrm{DTFT}\left\{x[n] * h[n]\right\} = ?$$

Intuition: the DTFT reconstruction formula tells us how to build $x[n]$ from a set of complex exponential "basis" functions. By linearity...

The convolution theorem

In general:

$$\text{DTFT}\left\{x[n] * h[n]\right\} = ?$$

Intuition: the DTFT reconstruction formula tells us how to build $x[n]$ from a set of complex exponential "basis" functions. By linearity...

The convolution theorem

$$\begin{aligned}
\text{DTFT}\{x[n] * h[n]\} &= \sum_{n=-\infty}^{\infty} (x * h)[n] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega(n-k)} e^{-j\omega k} \\
&= \sum_{k=-\infty}^{\infty} x[k] e^{-j\omega k} \sum_{n=-\infty}^{\infty} h[n-k] e^{-j\omega(n-k)} \\
&= H(e^{j\omega}) X(e^{j\omega})
\end{aligned}$$

The convolution theorem

$$
\begin{aligned}
\mathrm{DTFT}\{x[n] * h[n]\} &= \sum_{n=-\infty}^{\infty} (x * h)[n] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega (n-k)} e^{-j\omega k} \\
&= \sum_{k=-\infty}^{\infty} x[k] e^{-j\omega k} \sum_{n=-\infty}^{\infty} h[n-k] e^{-j\omega (n-k)} \\
&= H(e^{j\omega}) X(e^{j\omega})
\end{aligned}
$$

The convolution theorem

$$\begin{aligned}
\mathrm{DTFT}\{x[n] * h[n]\} &= \sum_{n=-\infty}^{\infty} (x * h)[n] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega(n-k)} e^{-j\omega k} \\
&= \sum_{k=-\infty}^{\infty} x[k] e^{-j\omega k} \sum_{n=-\infty}^{\infty} h[n-k] e^{-j\omega(n-k)} \\
&= H(e^{j\omega}) X(e^{j\omega})
\end{aligned}$$

The convolution theorem

$$
\begin{aligned}
\mathrm{DTFT}\left\{x[n] * h[n]\right\} &= \sum_{n=-\infty}^{\infty} (x * h)[n] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega(n-k)} e^{-j\omega k} \\
&= \sum_{k=-\infty}^{\infty} x[k] e^{-j\omega k} \sum_{n=-\infty}^{\infty} h[n-k] e^{-j\omega(n-k)} \\
&= H(e^{j\omega}) X(e^{j\omega})
\end{aligned}
$$

The convolution theorem

$$\begin{aligned}
\mathrm{DTFT}\left\{x[n] * h[n]\right\} &= \sum_{n=-\infty}^{\infty} (x * h)[n] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] h[n-k] e^{-j\omega(n-k)} e^{-j\omega k} \\
&= \sum_{k=-\infty}^{\infty} x[k] e^{-j\omega k} \sum_{n=-\infty}^{\infty} h[n-k] e^{-j\omega(n-k)} \\
&= H(e^{j\omega}) X(e^{j\omega})
\end{aligned}$$

Frequency response

$$H(e^{j\omega}) = \mathrm{DTFT}\left\{h[n]\right\}$$

Two effects:

- **magnitude:** amplification ($|H(e^{j\omega})| > 1$) or attenuation ($|H(e^{j\omega})| < 1$) of input frequencies
- **phase:** overall delay and shape changes

Frequency response

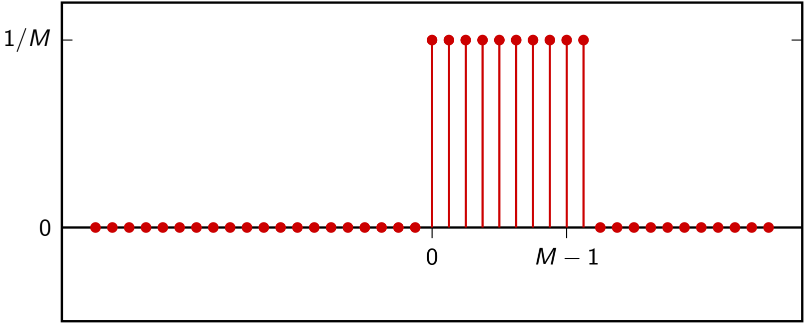
$$H(e^{j\omega}) = \text{DTFT}\{h[n]\}$$

Two effects:

- magnitude: amplification ($|H(e^{j\omega})| > 1$) or attenuation ($|H(e^{j\omega})| < 1$) of input frequencies
- phase: overall delay and shape changes

Moving Average revisited

$$h[n] = (u[n] - u[n-M])/M$$

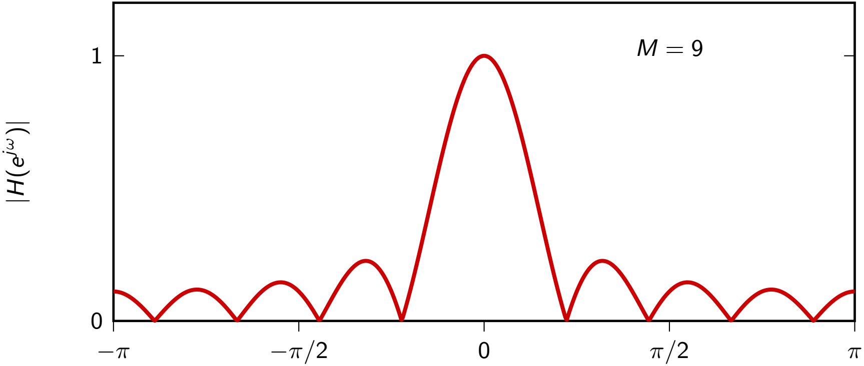


Moving Average revisited

$$
\begin{aligned}
H(e^{j\omega}) &= \sum_{n=0}^{M-1} \frac{1}{M}\, e^{-j\omega n} \\
&= \frac{1}{M}\, \frac{1 - e^{-j\omega M}}{1 - e^{-j\omega}} \\
&= \frac{1}{M}\, \frac{e^{-j\frac{\omega M}{2}}\left[e^{j\frac{\omega M}{2}} - e^{-j\frac{\omega M}{2}}\right]}{e^{-j\frac{\omega}{2}}\left[e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}\right]} \\
&= \frac{1}{M}\, \frac{\sin\left(\frac{\omega}{2}M\right)}{\sin\left(\frac{\omega}{2}\right)} e^{-j\frac{\omega}{2}(M-1)}
\end{aligned}
$$

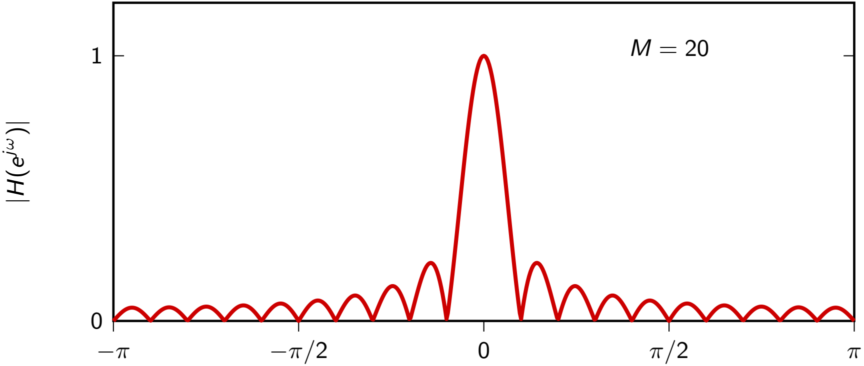
Moving Average, magnitude response

$$|H(e^{j\omega})| = \frac{1}{M}\left|\frac{\sin\left(\frac{\omega}{2}M\right)}{\sin\left(\frac{\omega}{2}\right)}\right|$$



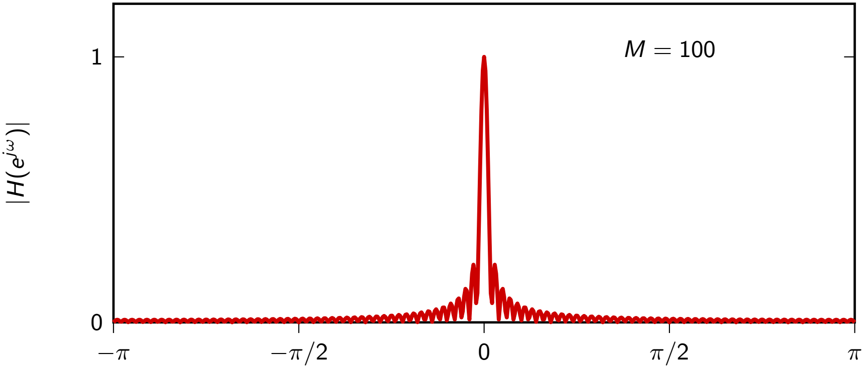
Moving Average, magnitude response

$$|H(e^{j\omega})| = \frac{1}{M}\left|\frac{\sin\left(\frac{\omega}{2}M\right)}{\sin\left(\frac{\omega}{2}\right)}\right|$$

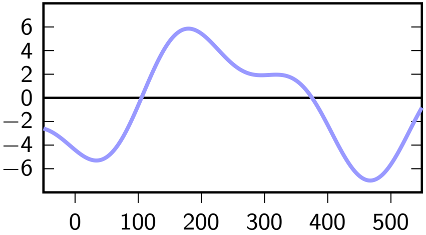


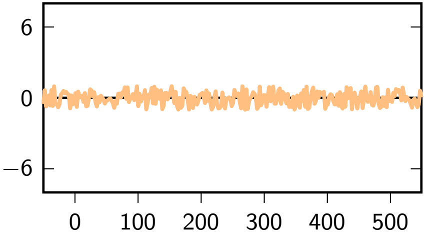
Moving Average, magnitude response

$$|H(e^{j\omega})| = \frac{1}{M} \left| \frac{\sin\left(\frac{\omega}{2} M\right)}{\sin\left(\frac{\omega}{2}\right)} \right|$$

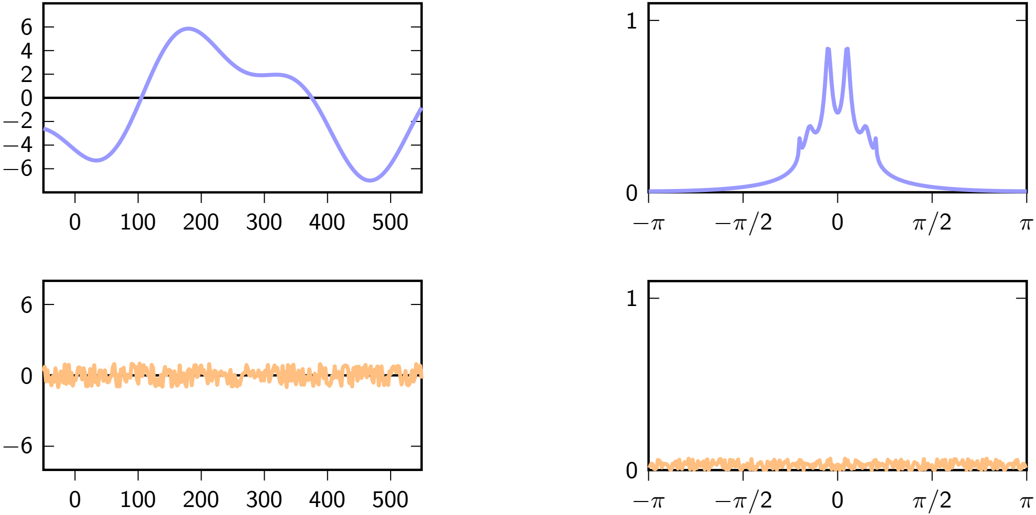


Denoising revisited

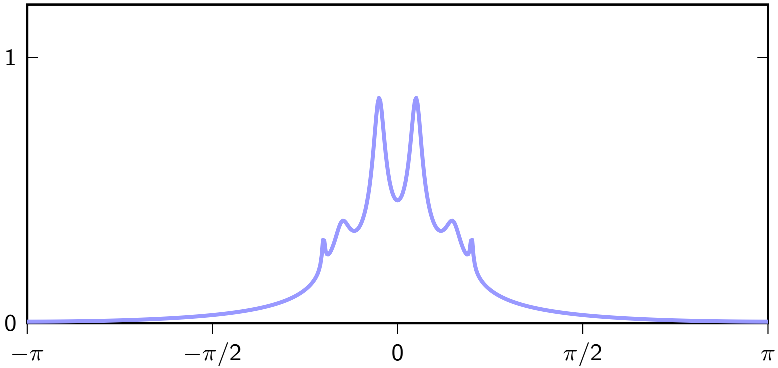




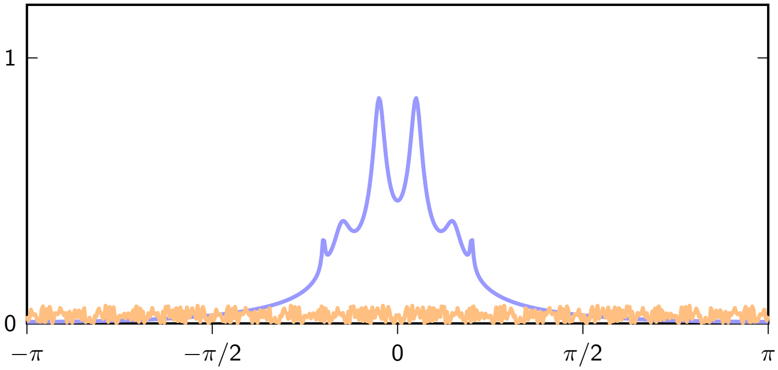
Denoising revisited



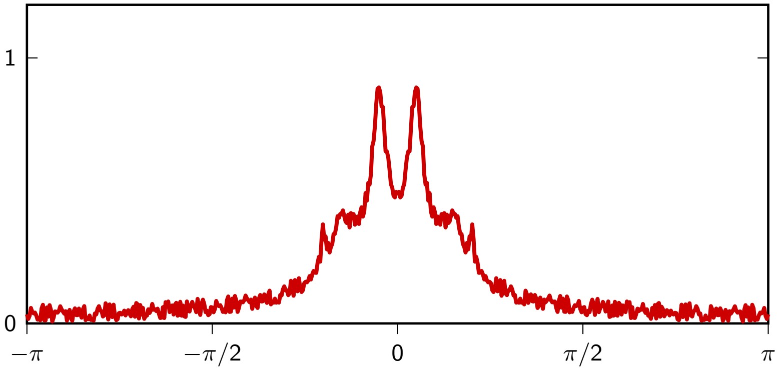
Denoising revisited



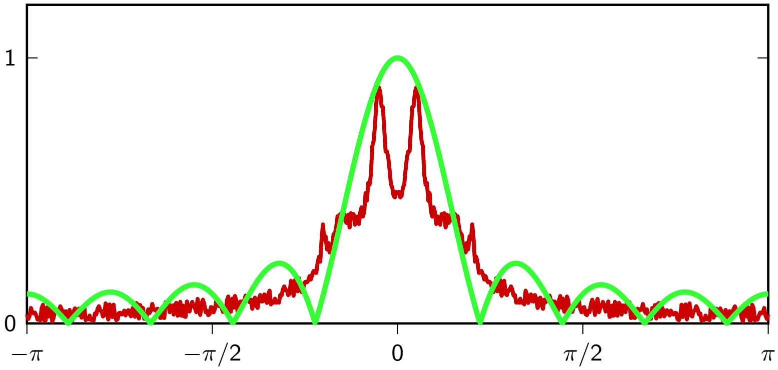
Denoising revisited



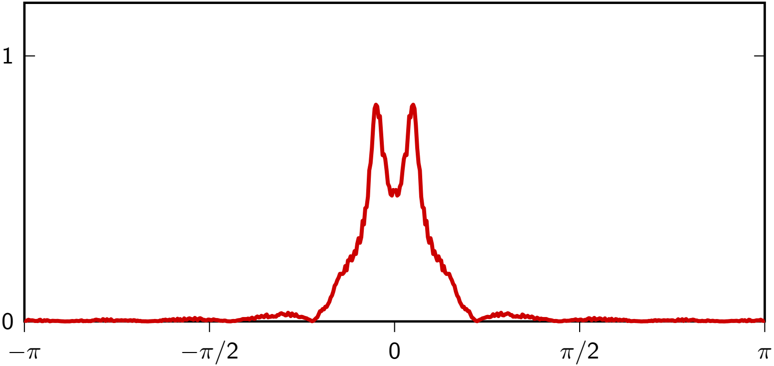
Denoising revisited



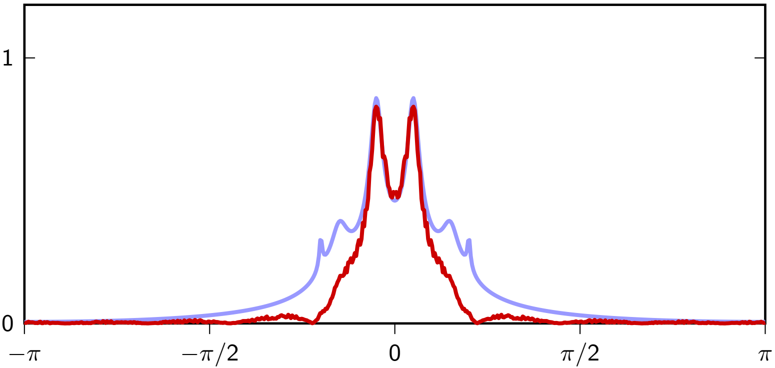
Denoising revisited



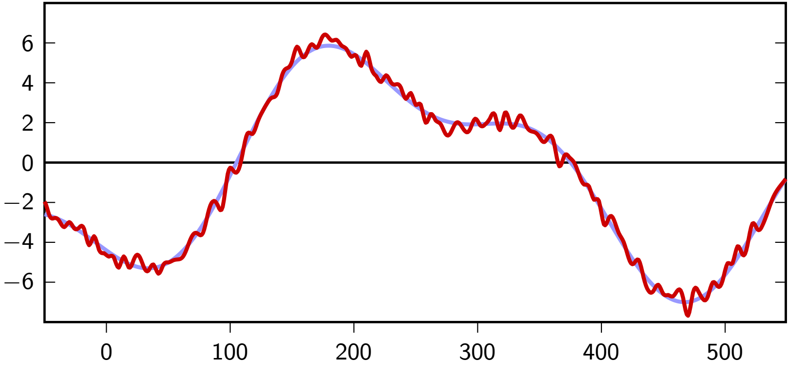
Denoising revisited



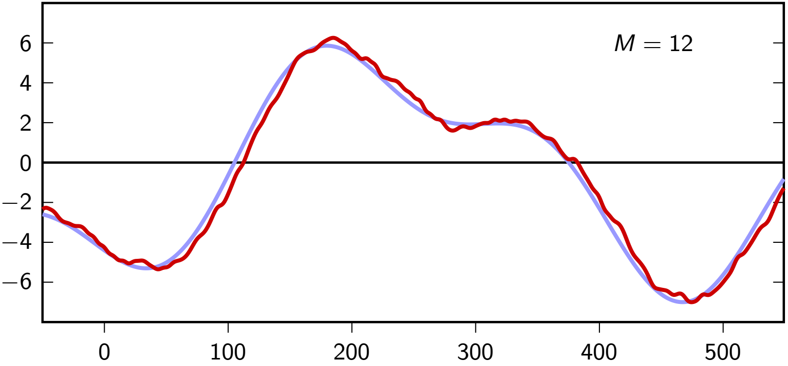
Denoising revisited



By the way, remember the time-domain analysis...



By the way, remember the time-domain analysis...



What about the phase?

Assume $|H(e^{j\omega})| = 1$

- zero phase: $\angle H(e^{j\omega}) = 0$
- linear phase: $\angle H(e^{j\omega}) = d\omega$
- nonlinear phase

What about the phase?

Assume $|H(e^{j\omega})| = 1$

- zero phase: $\angle H(e^{j\omega}) = 0$
- linear phase: $\angle H(e^{j\omega}) = d\omega$
- nonlinear phase

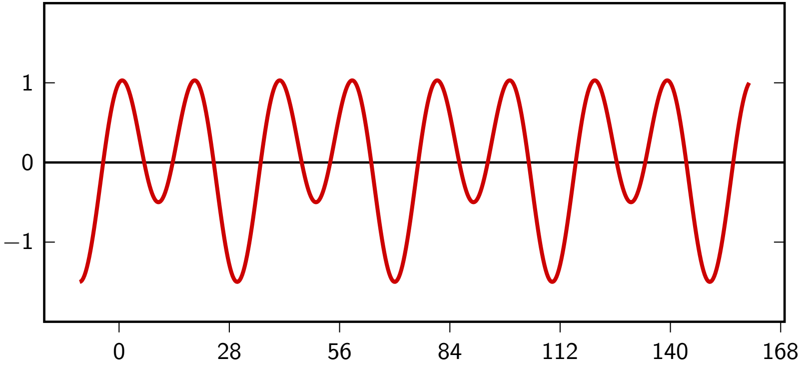
What about the phase?

Assume $|H(e^{j\omega})| = 1$

- zero phase: $\angle H(e^{j\omega}) = 0$
- linear phase: $\angle H(e^{j\omega}) = d\omega$
- nonlinear phase

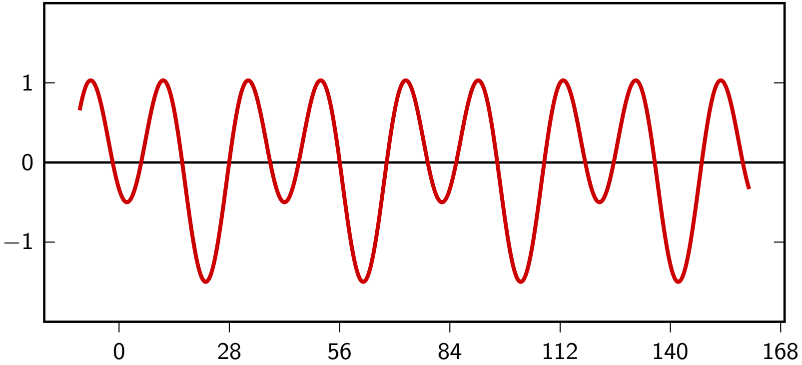
Phase and signal shape

$$x[n] = \frac{1}{2}\sin(\omega_0 n) + \cos(2\omega_0 n) \qquad \omega_0 = \frac{2\pi}{40}$$



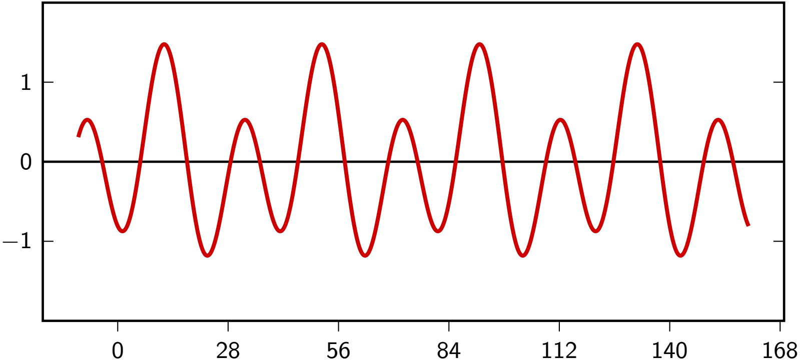
Phase and signal shape: linear phase

$$x[n] = \frac{1}{2}\sin(\omega_0 n + \theta_0) + \cos(2\omega_0 n + 2\theta_0) \qquad \theta_0 = \frac{8\pi}{5}$$



Phase and signal shape: nonlinear phase

$$x[n] = \frac{1}{2}\sin(\omega_0 n) + \cos(2\omega_0 n + 2\theta_0)$$

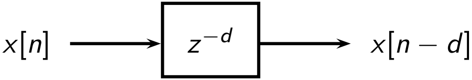


Linear phase

$$x[n] \longrightarrow \boxed{z^{-d}} \longrightarrow x[n-d]$$

- $y[n] = x[n-d]$
- $Y(e^{j\omega}) = e^{-j\omega d}\, X(e^{j\omega})$
- $H(e^{j\omega}) = e^{-j\omega d}$
- linear phase term

Linear phase



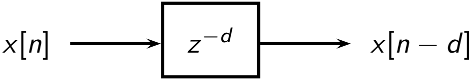
- $y[n] = x[n-d]$
- $Y(e^{j\omega}) = e^{-j\omega d}\, X(e^{j\omega})$
- $H(e^{j\omega}) = e^{-j\omega d}$
- linear phase term

Linear phase

$$x[n] \longrightarrow \boxed{z^{-d}} \longrightarrow x[n-d]$$

- $y[n] = x[n-d]$
- $Y(e^{j\omega}) = e^{-j\omega d}\, X(e^{j\omega})$
- $H(e^{j\omega}) = e^{-j\omega d}$
- linear phase term

Linear phase



- $y[n] = x[n-d]$
- $Y(e^{j\omega}) = e^{-j\omega d} X(e^{j\omega})$
- $H(e^{j\omega}) = e^{-j\omega d}$
- linear phase term

Linear phase

In general, if $H(e^{j\omega}) = A(e^{j\omega})e^{-j\omega d}, \quad$ with $A(e^{j\omega}) \in \mathbb{R}$

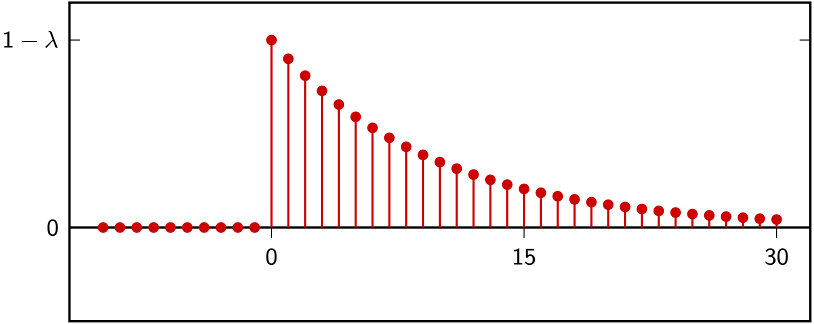
$$x[n] \longrightarrow \boxed{A(e^{j\omega})} \longrightarrow \boxed{\text{delay by } d} \longrightarrow y[n]$$

Moving Average is linear phase

$$H(e^{j\omega}) = \frac{1}{M}\,\frac{\sin\left(\frac{\omega}{2}M\right)}{\sin\left(\frac{\omega}{2}\right)}e^{-j\frac{M-1}{2}\omega}$$

Leaky integrator revisited

$$h[n] = (1-\lambda)\lambda^n \, u[n]$$



Leaky integrator revisited

$$H(e^{j\omega}) = \frac{1-\lambda}{1-\lambda e^{-j\omega}}$$

Finding magnitude and phase require a little algebra...

Leaky integrator revisited

Recall from complex algebra:

$$\frac{1}{a+jb} = \frac{a-jb}{a^2+b^2}$$

so that if $x = 1/(a+jb)$,

$$|x|^2 = \frac{1}{a^2+b^2}$$

$$\angle x = \tan^{-1}\left[-\frac{b}{a}\right]$$

Leaky integrator revisited

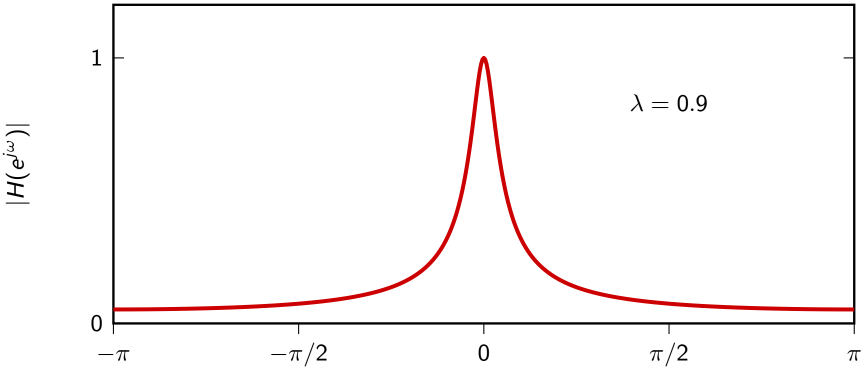
$$H(e^{j\omega}) = \frac{1-\lambda}{(1-\lambda\cos\omega) - j\lambda\sin\omega}$$

so that:

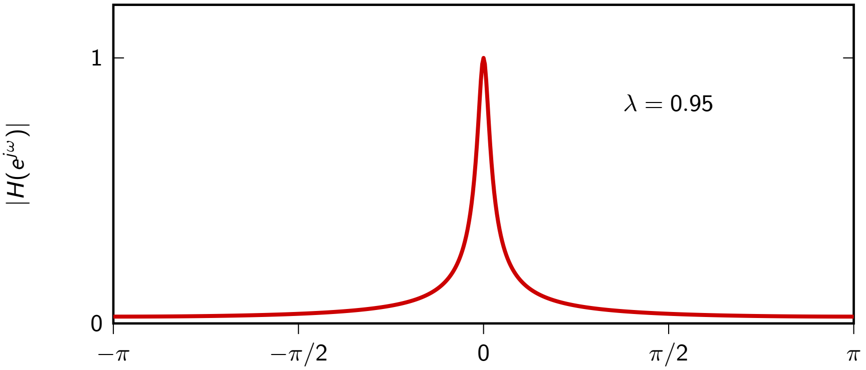
$$|H(e^{j\omega})|^2 = \frac{(1-\lambda)^2}{1-2\lambda\cos\omega+\lambda^2}$$

$$\angle H(e^{j\omega}) = \tan^{-1}\left[\frac{\lambda\sin\omega}{1-\lambda\cos\omega}\right]$$

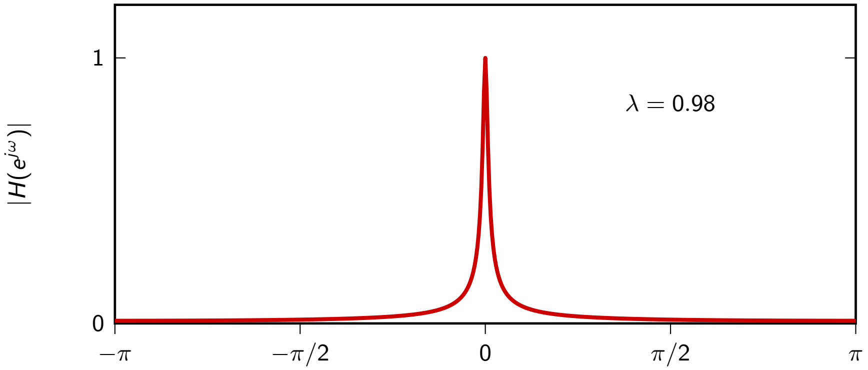
Leaky integrator, magnitude response



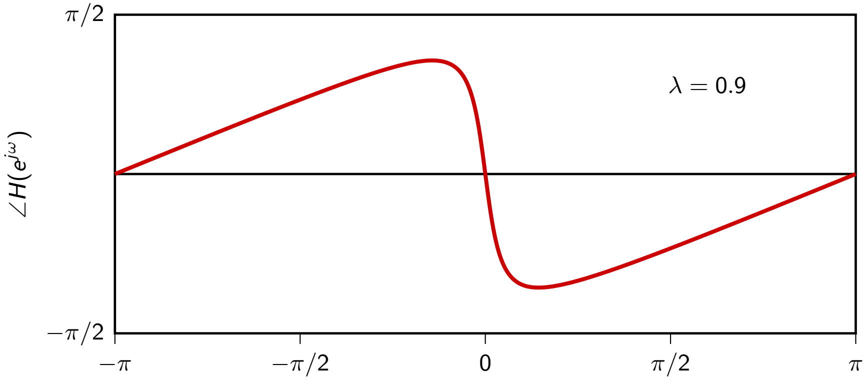
Leaky integrator, magnitude response



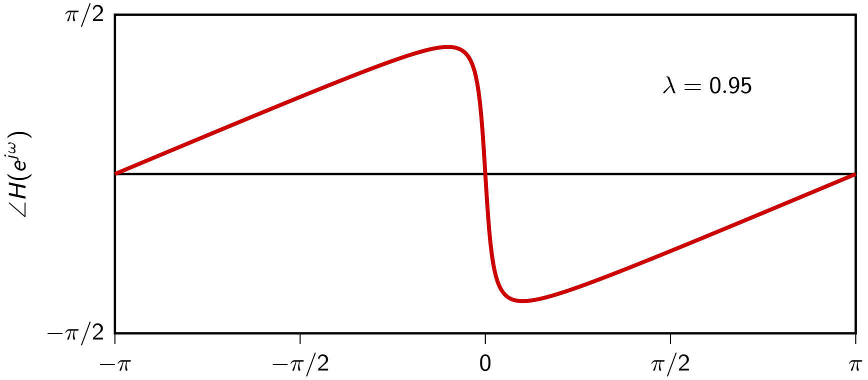
Leaky integrator, magnitude response



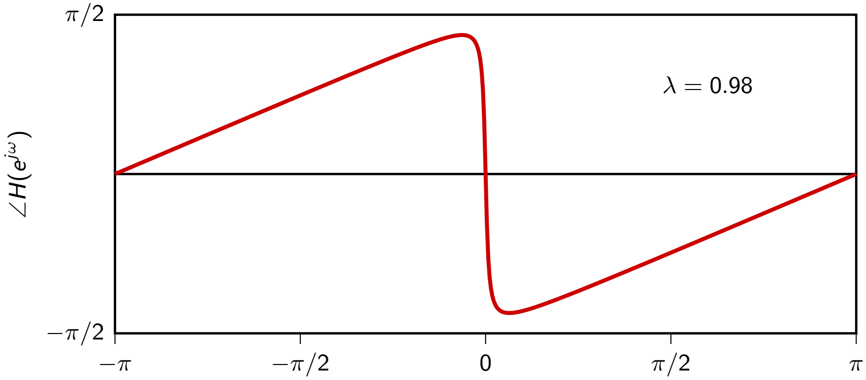
Leaky integrator, phase response



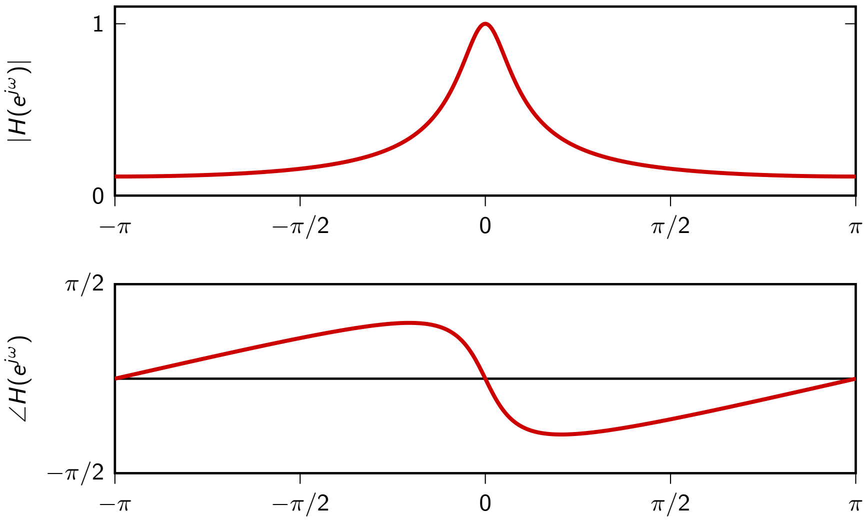
Leaky integrator, phase response



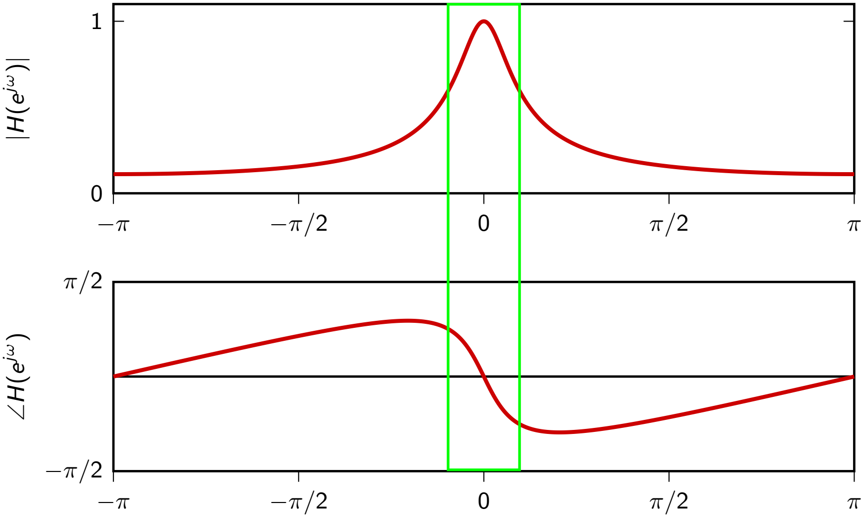
Leaky integrator, phase response



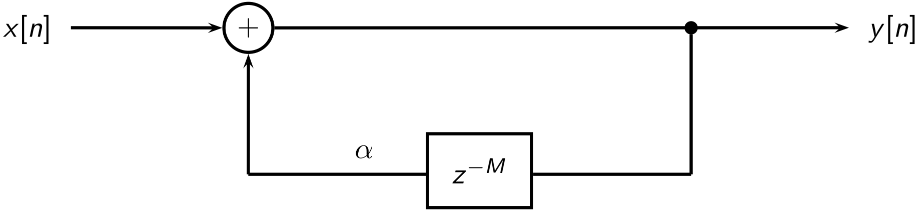
Phase is sufficiently linear where it matters



Phase is sufficiently linear where it matters



Karplus-Strong revisited, again!



$$y[n] = \alpha\, y[n-M] + x[n]$$

Karplus-Strong revisited

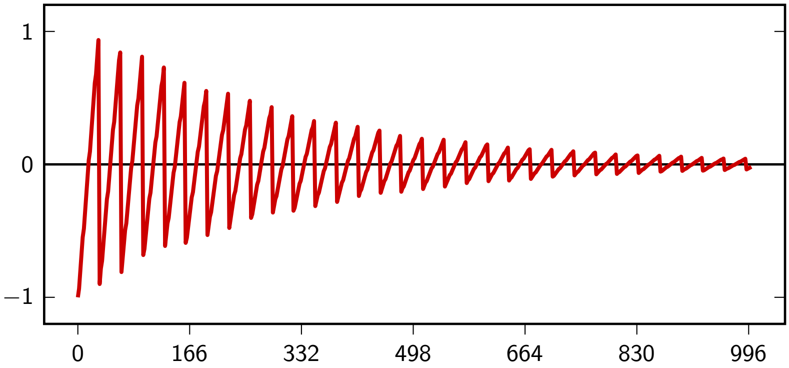
$$y[n] = \bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1], \alpha\bar{x}[0], \alpha\bar{x}[1], \ldots, \alpha\bar{x}[M-1], \alpha^2\bar{x}[0], \alpha^2\bar{x}[1], \ldots$$

Karplus-Strong revisited

$$y[n] = \underbrace{\bar{x}[0], \bar{x}[1], \ldots, \bar{x}[M-1]}_{1^{\text{st}}\text{ period}}, \underbrace{\alpha\bar{x}[0], \alpha\bar{x}[1], \ldots, \alpha\bar{x}[M-1]}_{2^{\text{nd}}\text{ period}}, \underbrace{\alpha^2\bar{x}[0], \alpha^2\bar{x}[1], \ldots}_{\ldots}$$

KS revisited: sawtooth signal

$$y[n] = \alpha^{\lfloor n/M \rfloor} \bar{x}[n \mod M]\, u[n]$$



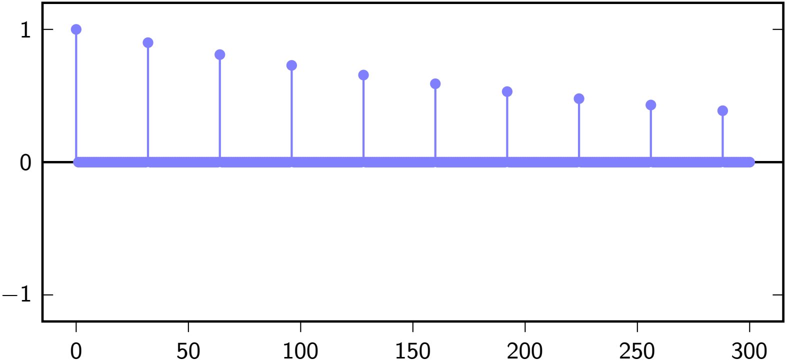
DTFT of KS signal, using the convolution theorem

key observation:

$$y[n] = \bar{x}[n] * w[n], \qquad w[n] = \begin{cases} \alpha^k & \text{for } n = kM \\ 0 & \text{otherwise} \end{cases}$$

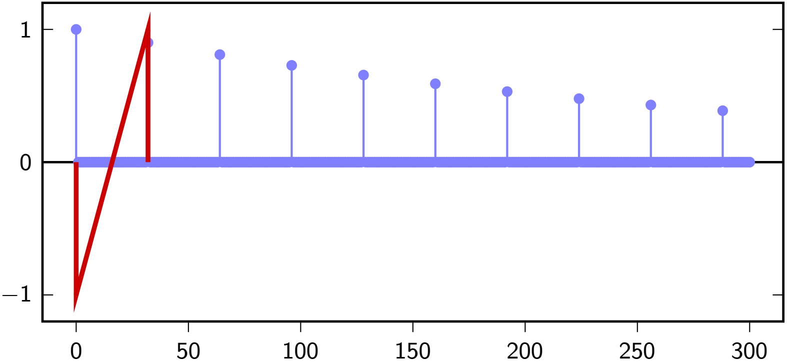
KS revisited

$$y[n] = \alpha^{\lfloor n/M \rfloor} \bar{x}[n \mod M]\, u[n]$$



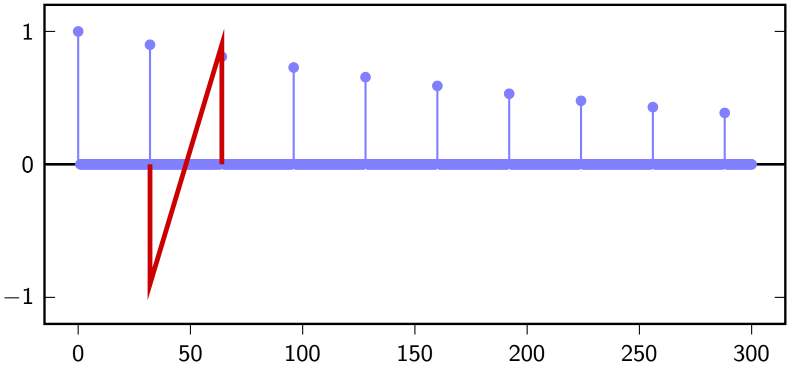
KS revisited

$$y[n] = \alpha^{\lfloor n/M \rfloor} \bar{x}[n \mod M]\, u[n]$$



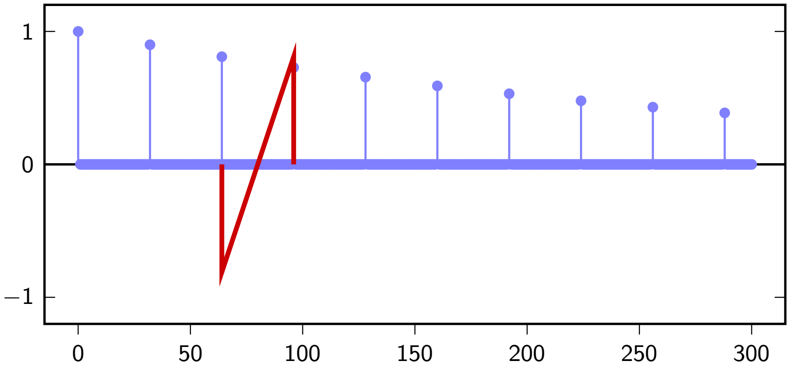
KS revisited

$$y[n] = \alpha^{\lfloor n/M \rfloor} \bar{x}[n \mod M]\, u[n]$$



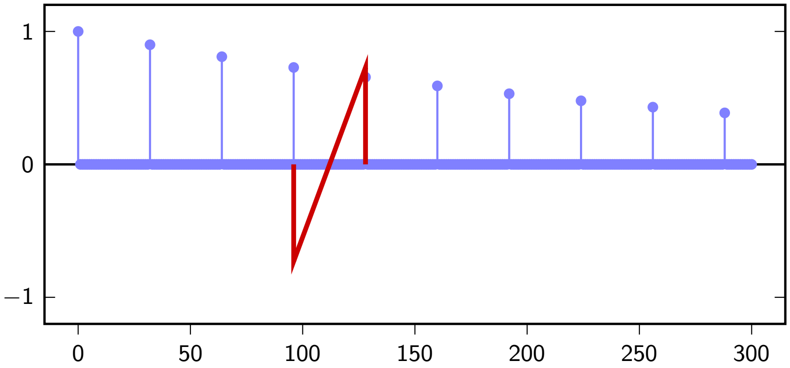
KS revisited

$$y[n] = \alpha^{\lfloor n/M \rfloor} \bar{x}[n \mod M]\, u[n]$$



KS revisited

$$y[n] = \alpha^{\lfloor n/M \rfloor} \bar{x}[n \mod M]\, u[n]$$



DTFT of KS signal, using the convolution theorem

key observation:

$$y[n] = \bar{x}[n] * w[n], \qquad w[n] = \begin{cases} \alpha^k & \text{for } n = kM \\ 0 & \text{otherwise} \end{cases}$$

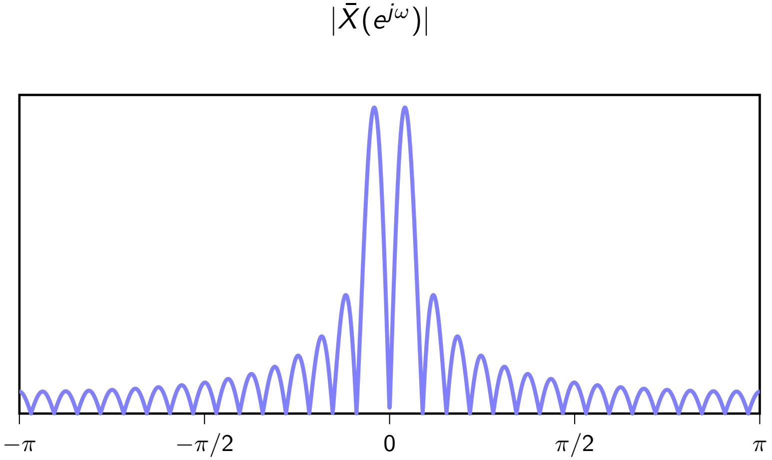
$$Y(e^{j\omega}) = \bar{X}(e^{j\omega})W(e^{j\omega})$$

DTFT of KS signal, using the convolution theorem

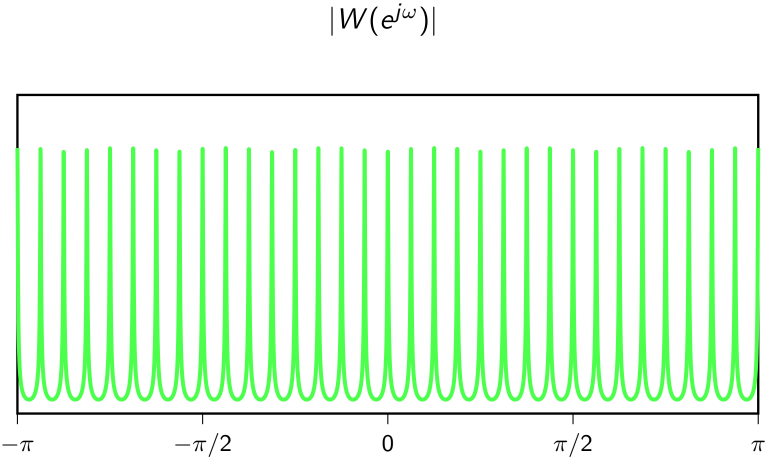
$$\bar{X}(e^{j\omega}) = e^{-j\omega}\left(\frac{M+1}{M-1}\right)\frac{1-e^{-j(M-1)\omega}}{(1-e^{-j\omega})^2} - \frac{1-e^{-j(M+1)\omega}}{(1-e^{-j\omega})^2}$$

$$W(e^{j\omega}) = \frac{1}{1-\alpha e^{-j\omega M}}$$

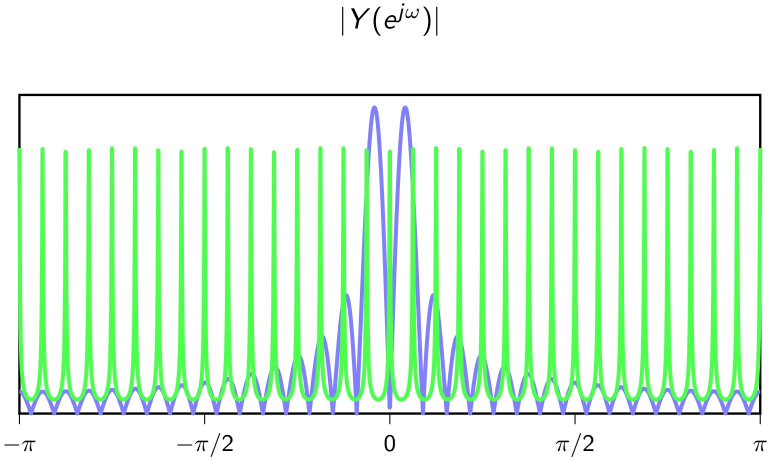
DTFT of KS



DTFT of KS



DTFT of KS



DTFT of KS