



Untitled, 2016

Tkacik Natural Image database convolved with Gaussian and Hadamard multiplied by block random matrices in MATLAB. In memory of Ellsworth Kelly.

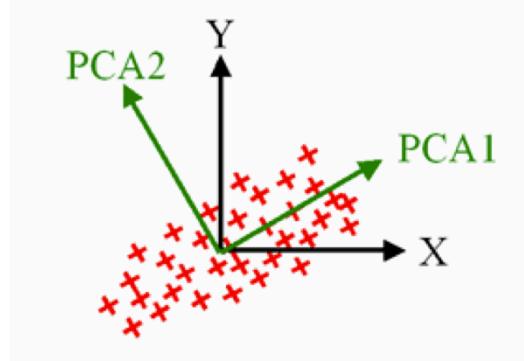
Lecture 3: Fourier Transforms, Filtering, and Convolutions

January 19th, 2016

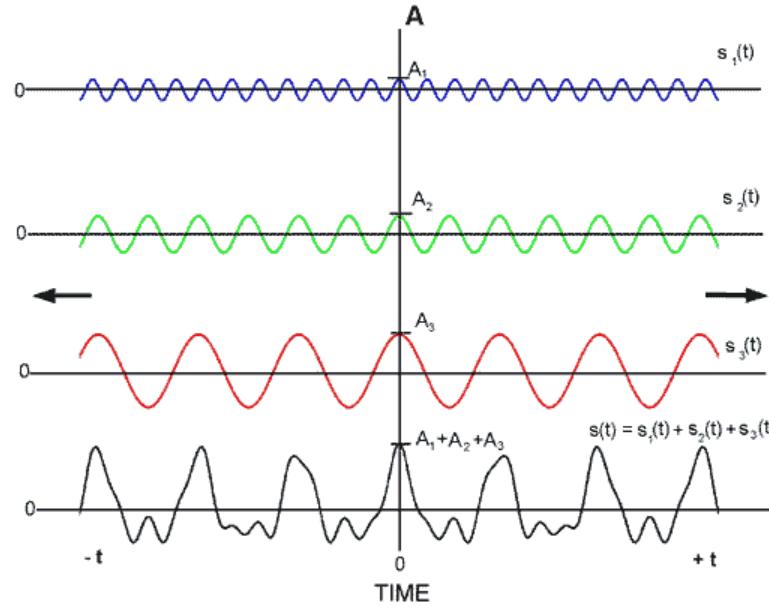
Lane McIntosh & Kiah Hardcastle

Math Tools for Neuroscience

Topics we will cover this quarter



dimensionality reduction

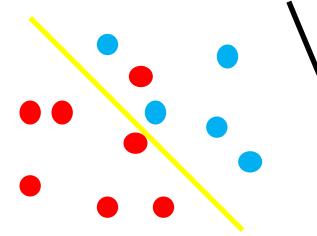


Fourier transforms, convolutions,
and filtering out noise

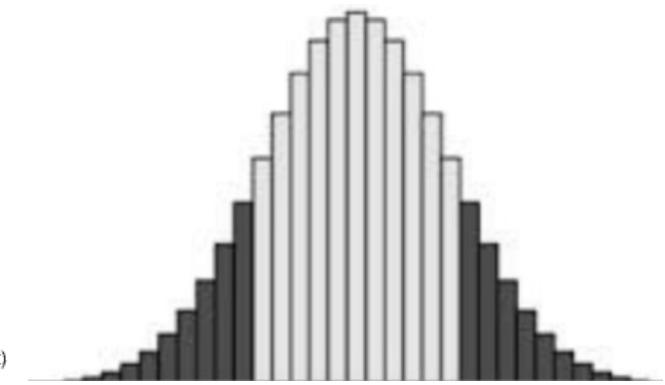
$$C_m \frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T e^{\frac{V-V_T}{\Delta T}} - u + I$$

$$\tau_w \frac{du}{dt} = a(V - E_L) - u$$

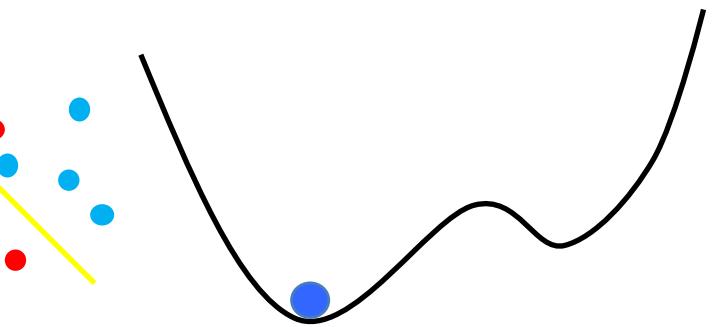
differential equations and modeling



statistical
models



statistics, Bayesian
probability, and
information theory



optimization

Today's lecture

When will this be useful?

Filtering in time or space

- What is convolution?
- How much should I filter?

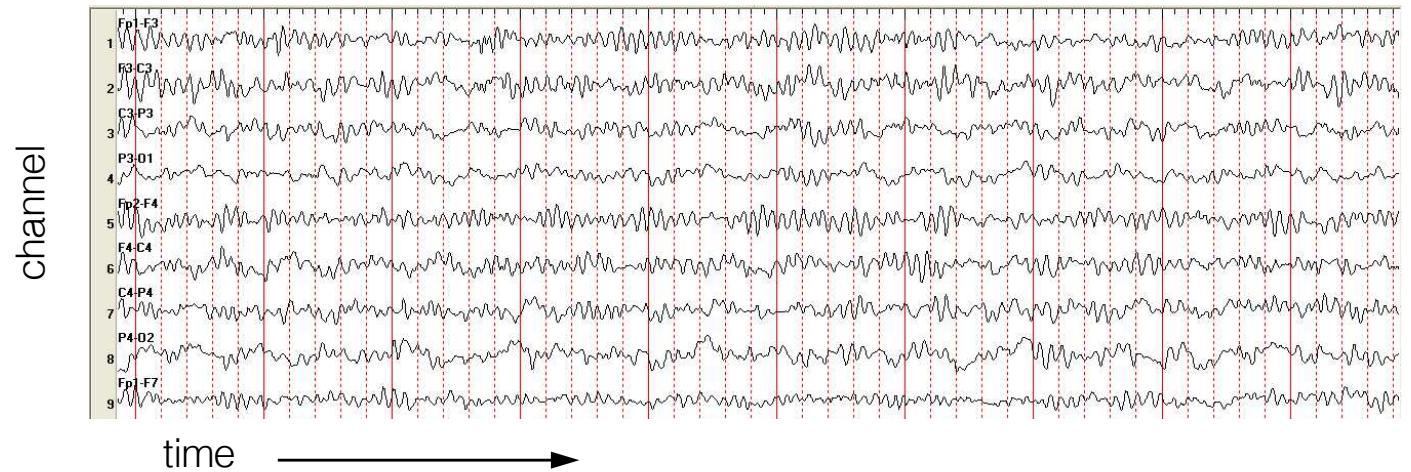
Filtering in frequency

- Fourier series
- What is a Fourier transform?
- Designing filters in frequency domain

Experimental examples of filtering

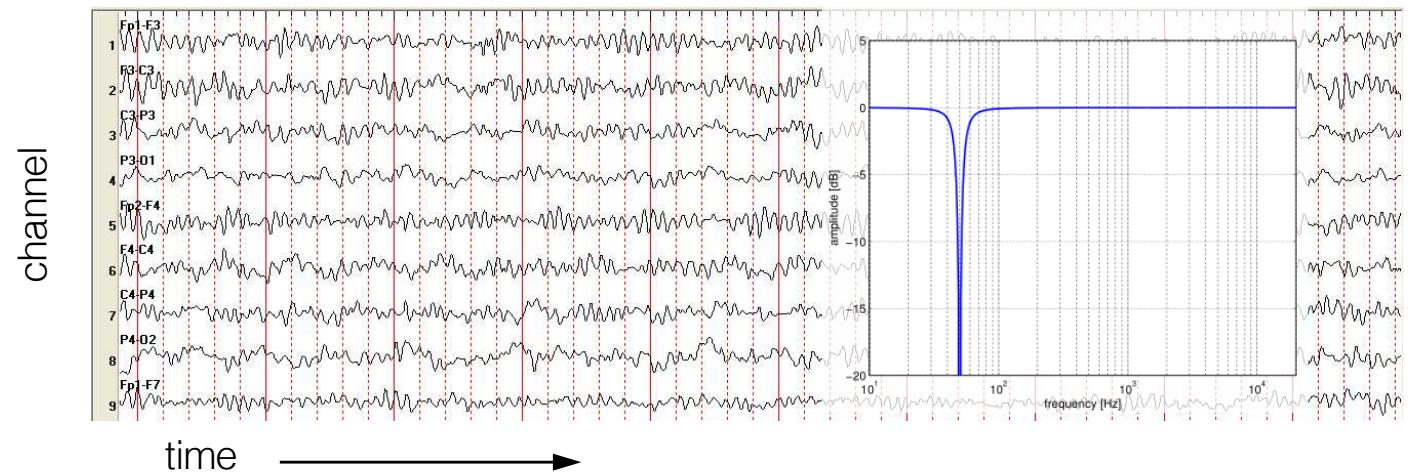
Experimental examples of filtering

Experimental noise



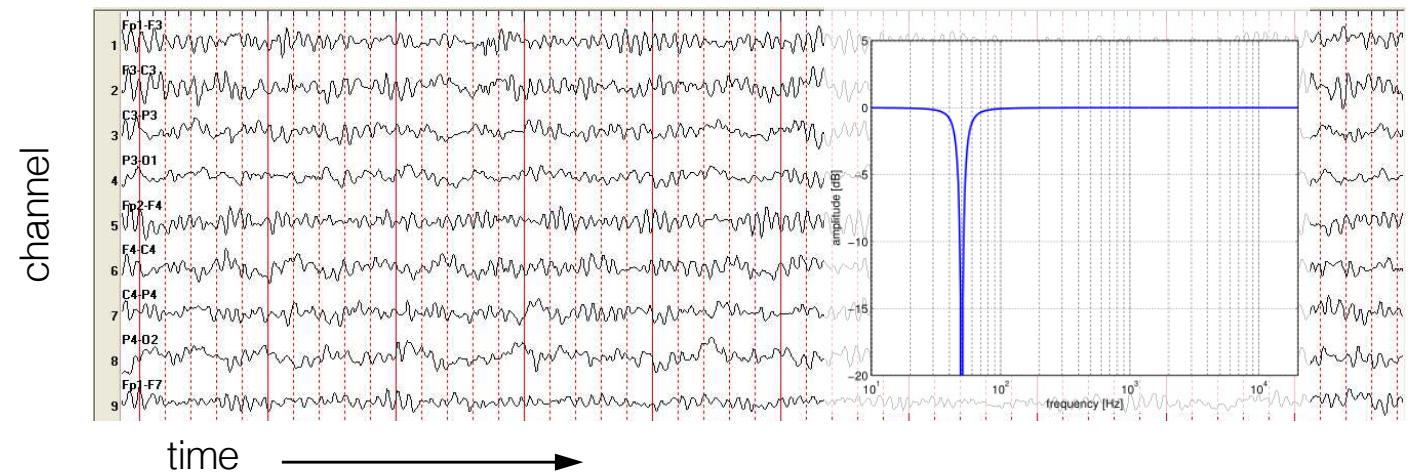
Experimental examples of filtering

Experimental noise

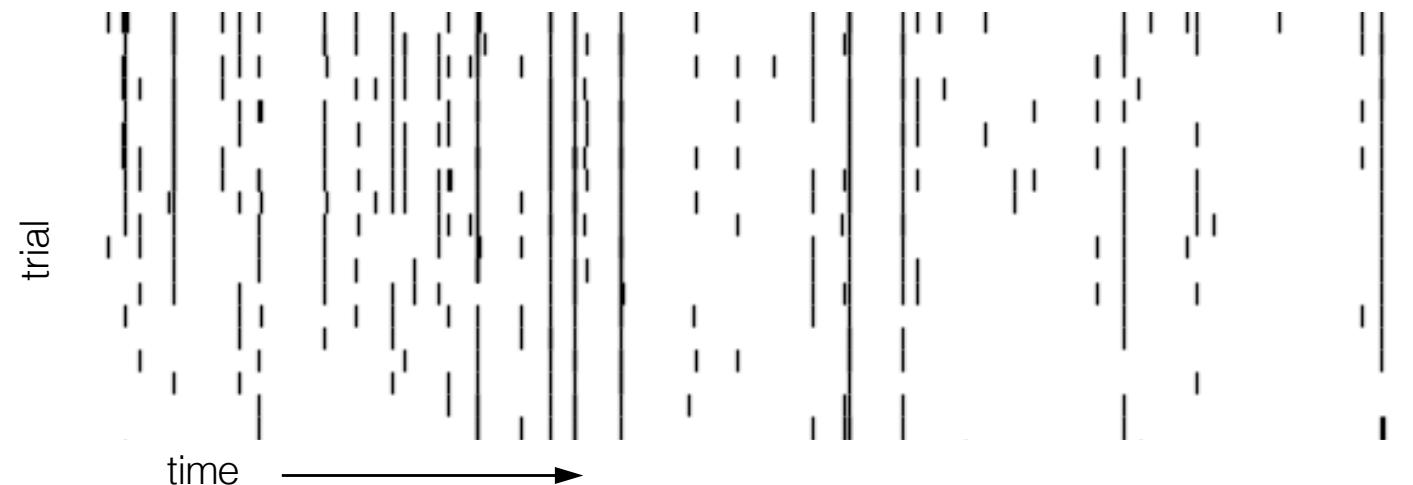
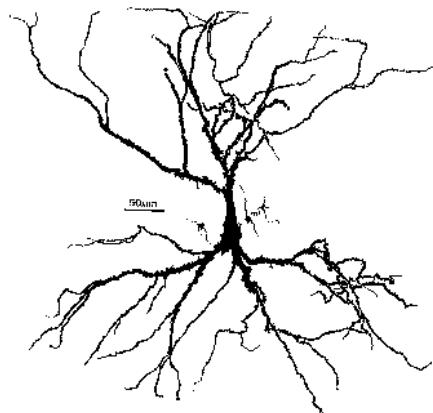


Experimental examples of filtering

Experimental noise

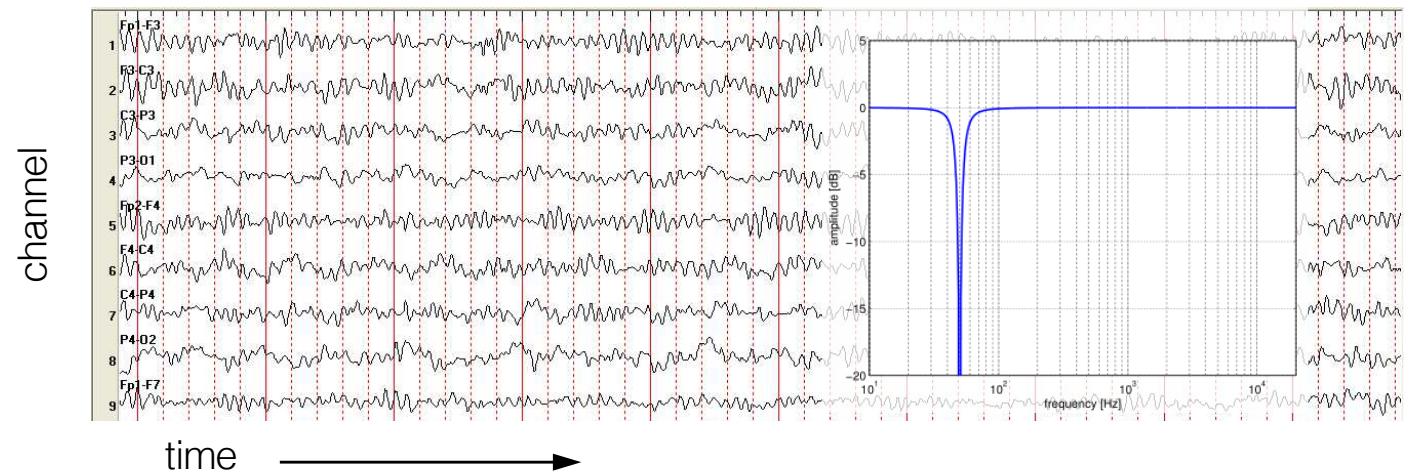


Biological noise

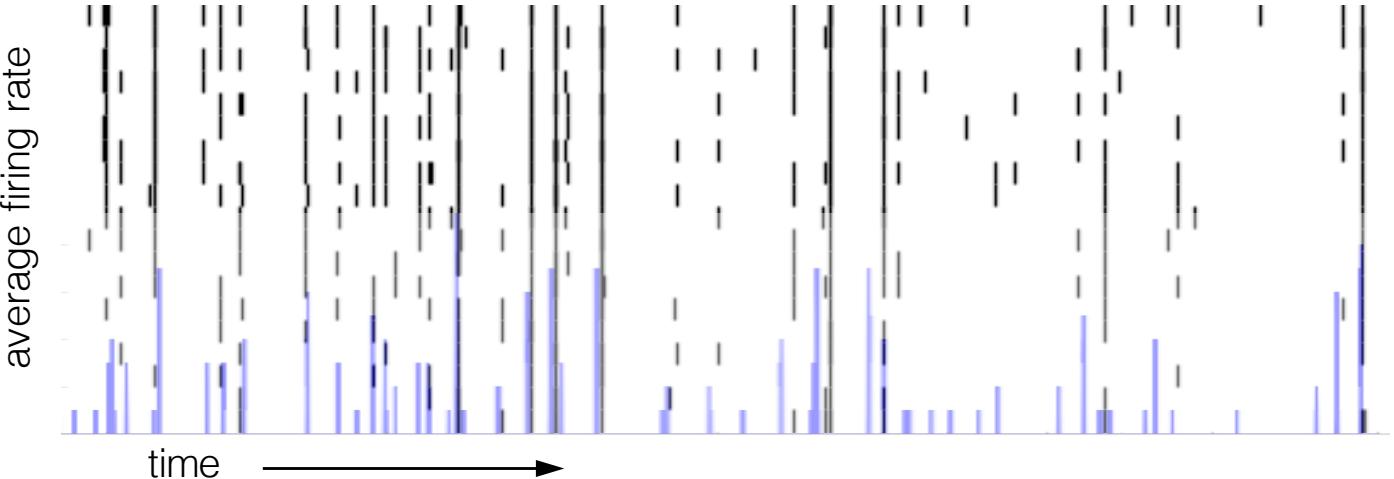
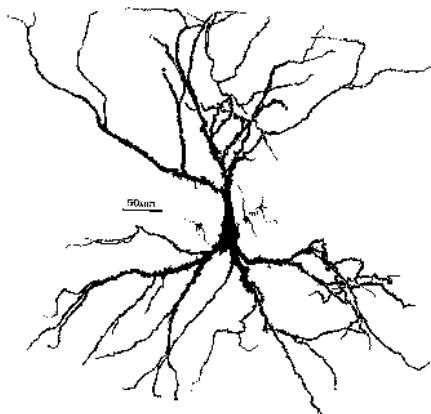


Experimental examples of filtering

Experimental noise

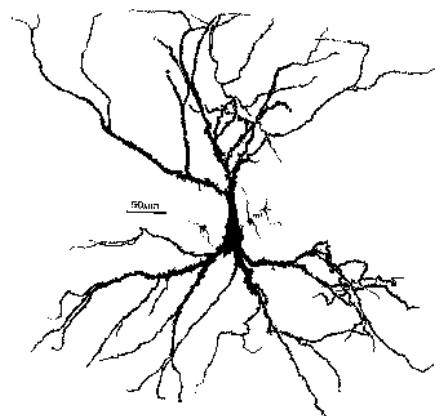
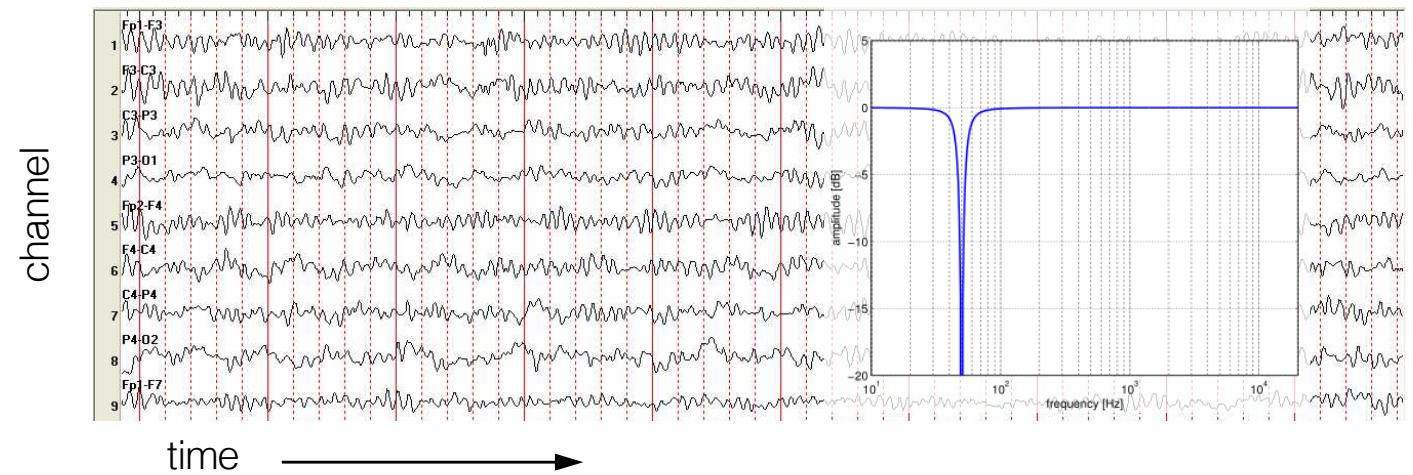


Biological noise

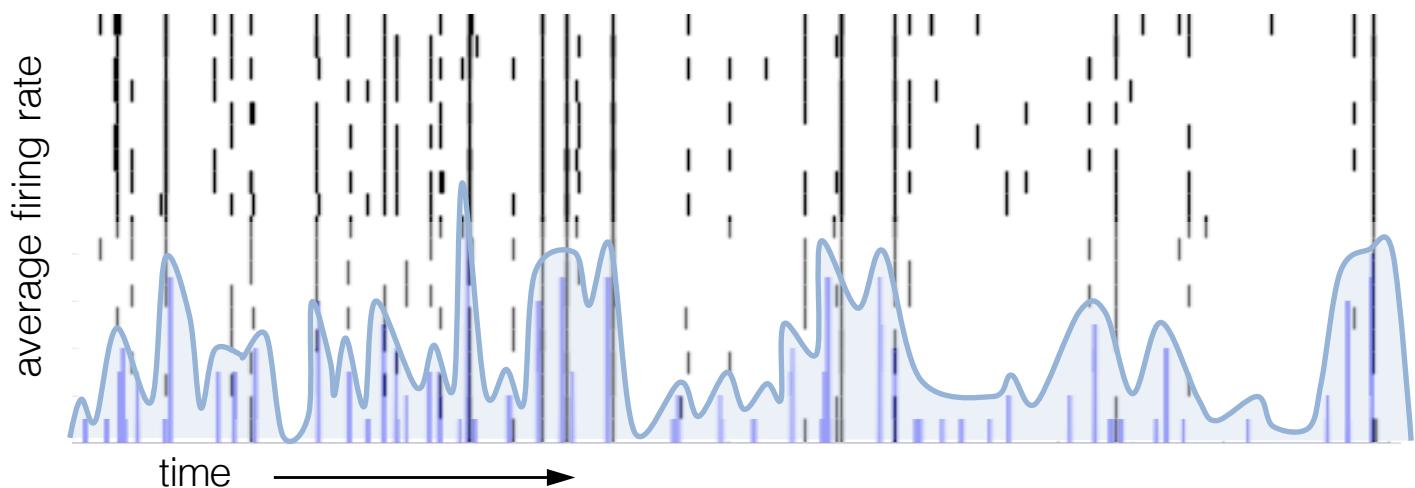


Experimental examples of filtering

Experimental noise



Biological noise



Today's lecture

When will this be useful?

Filtering in time or space

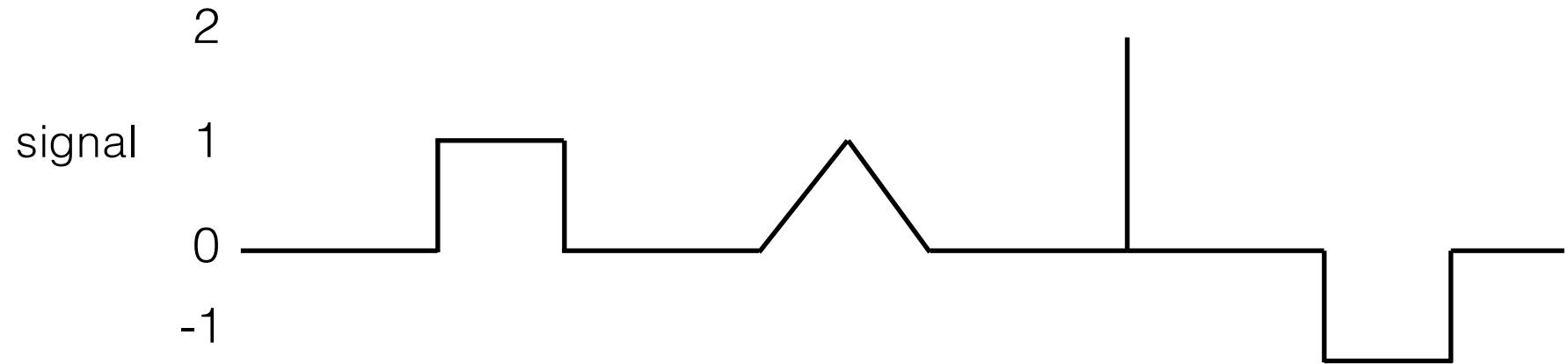
- What is convolution?
- How much should I filter?

Filtering in frequency

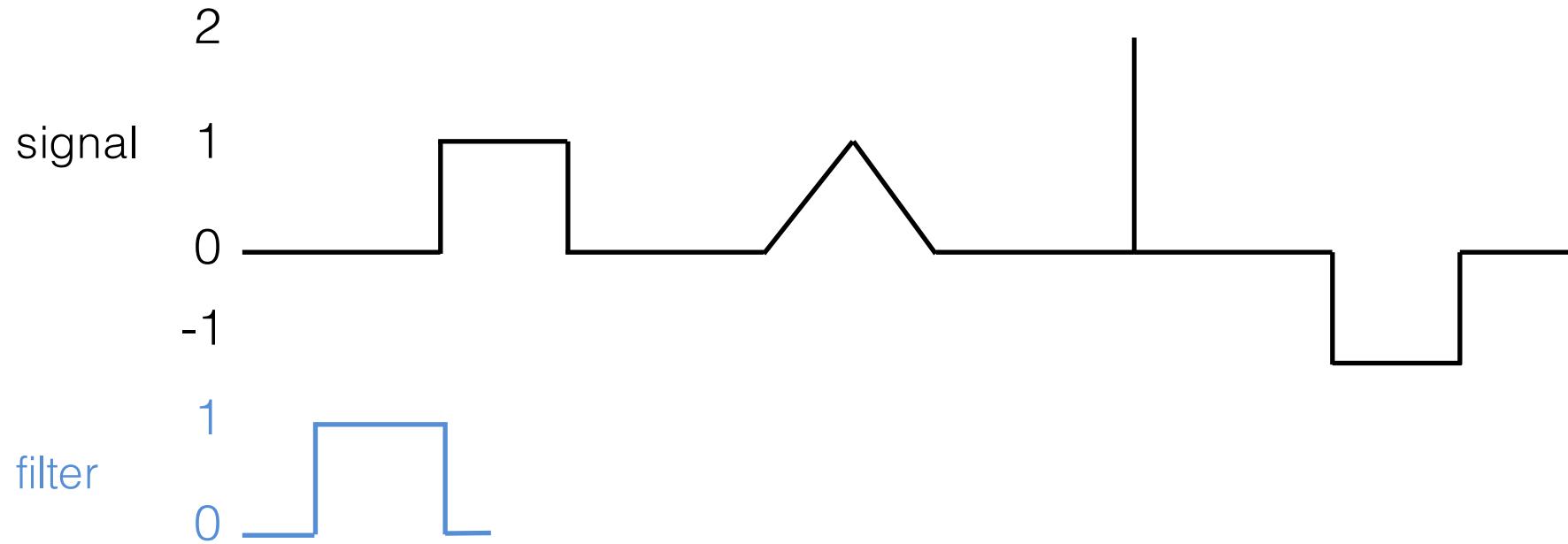
- Fourier series
- What is a Fourier transform?
- Designing filters in frequency domain

Convolution

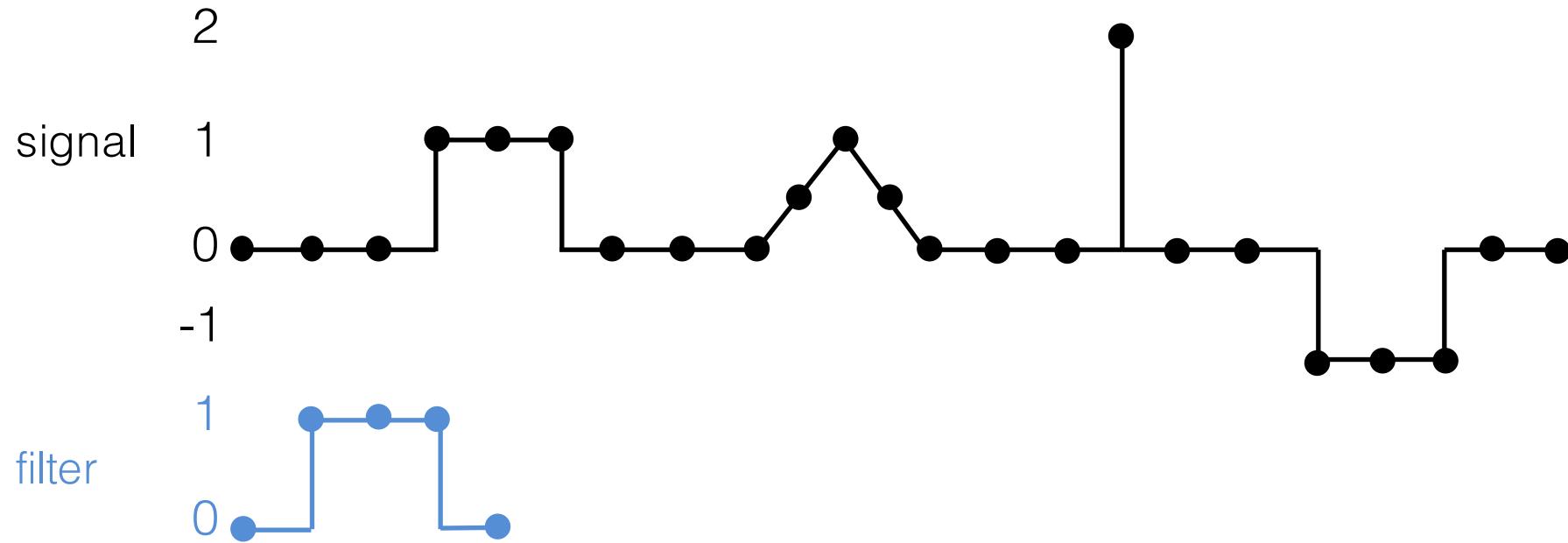
Convolution



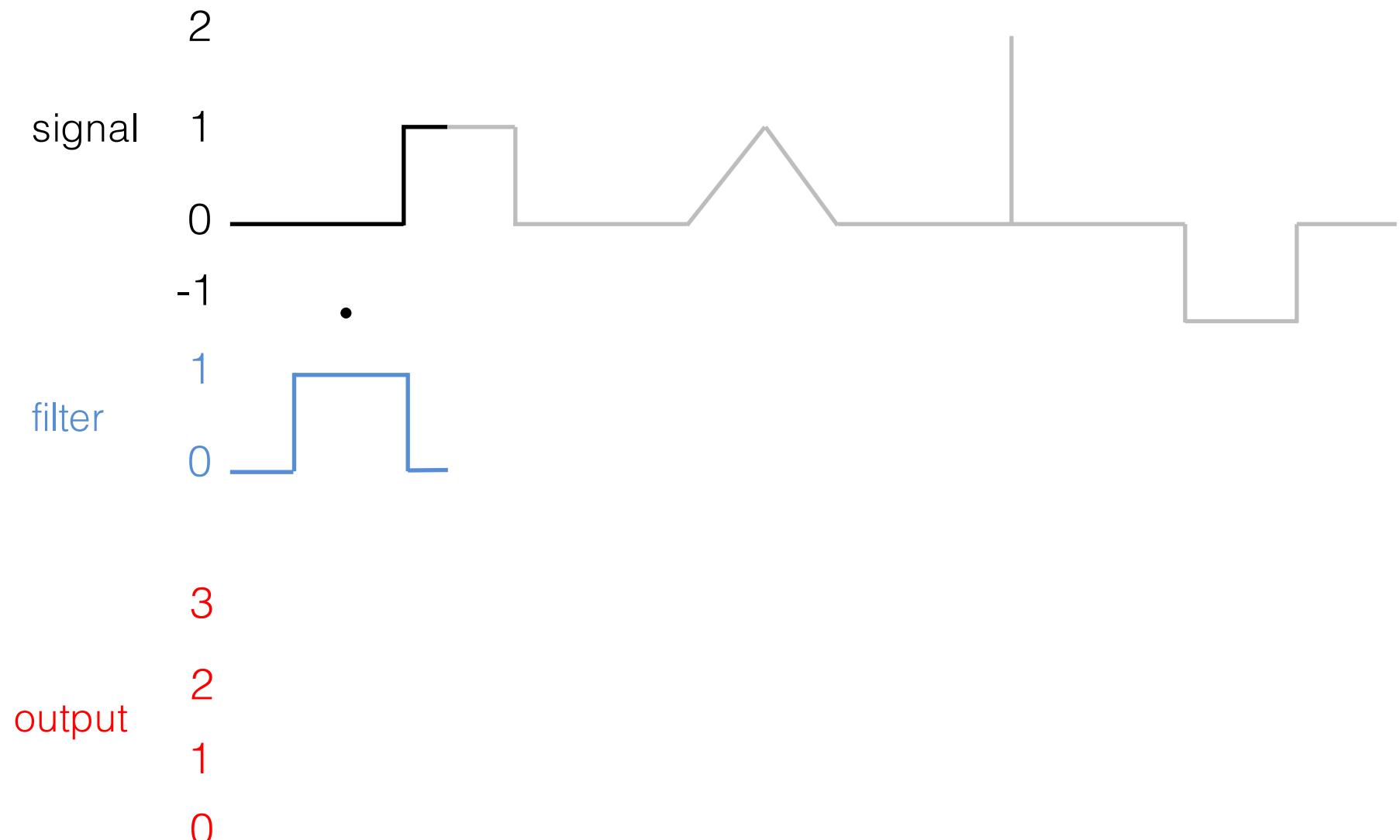
Convolution



Convolution

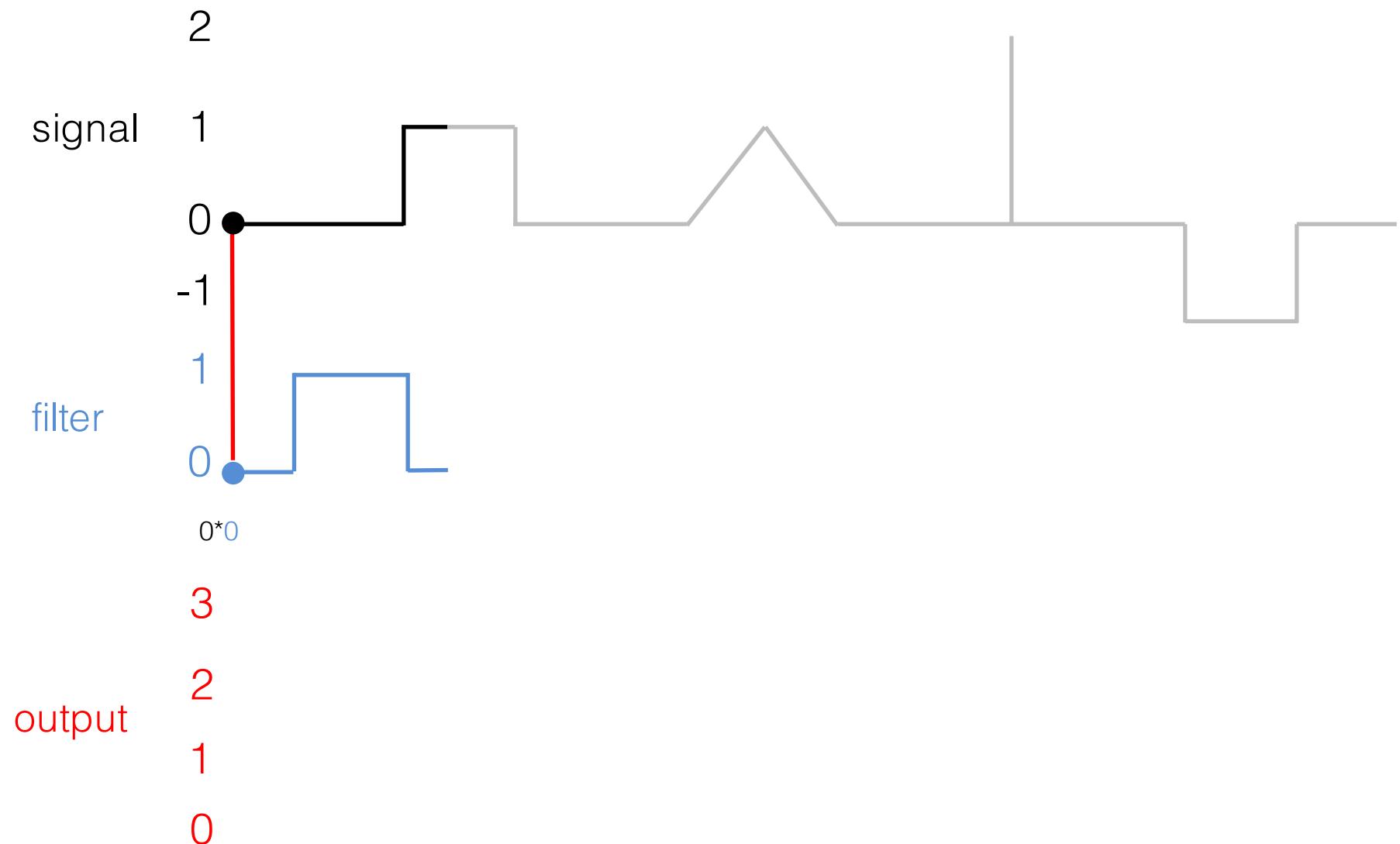


Convolution



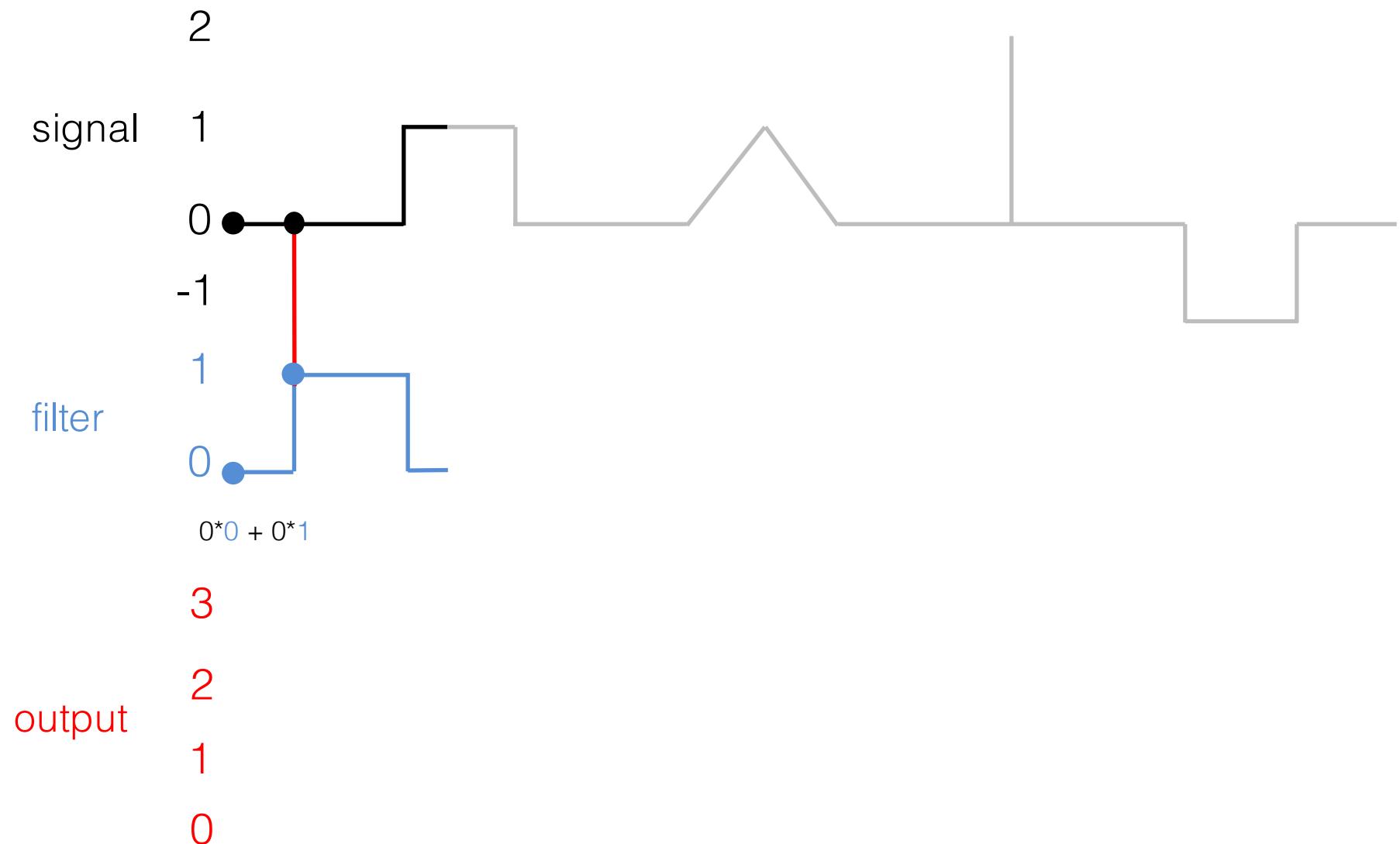
MATLAB: `conv(signal, filter, 'valid')`

Convolution



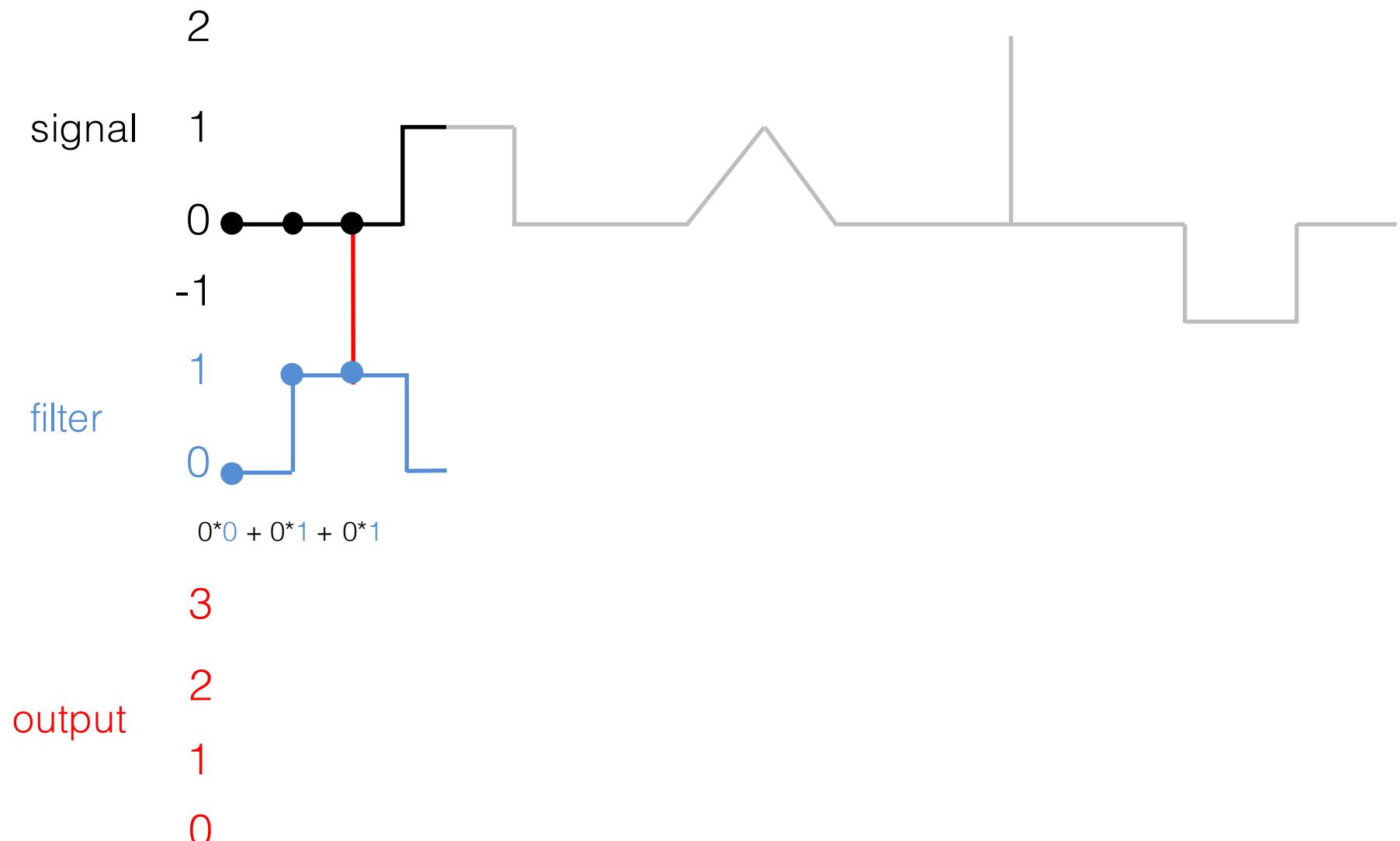
MATLAB: `conv(signal, filter, 'valid')`

Convolution

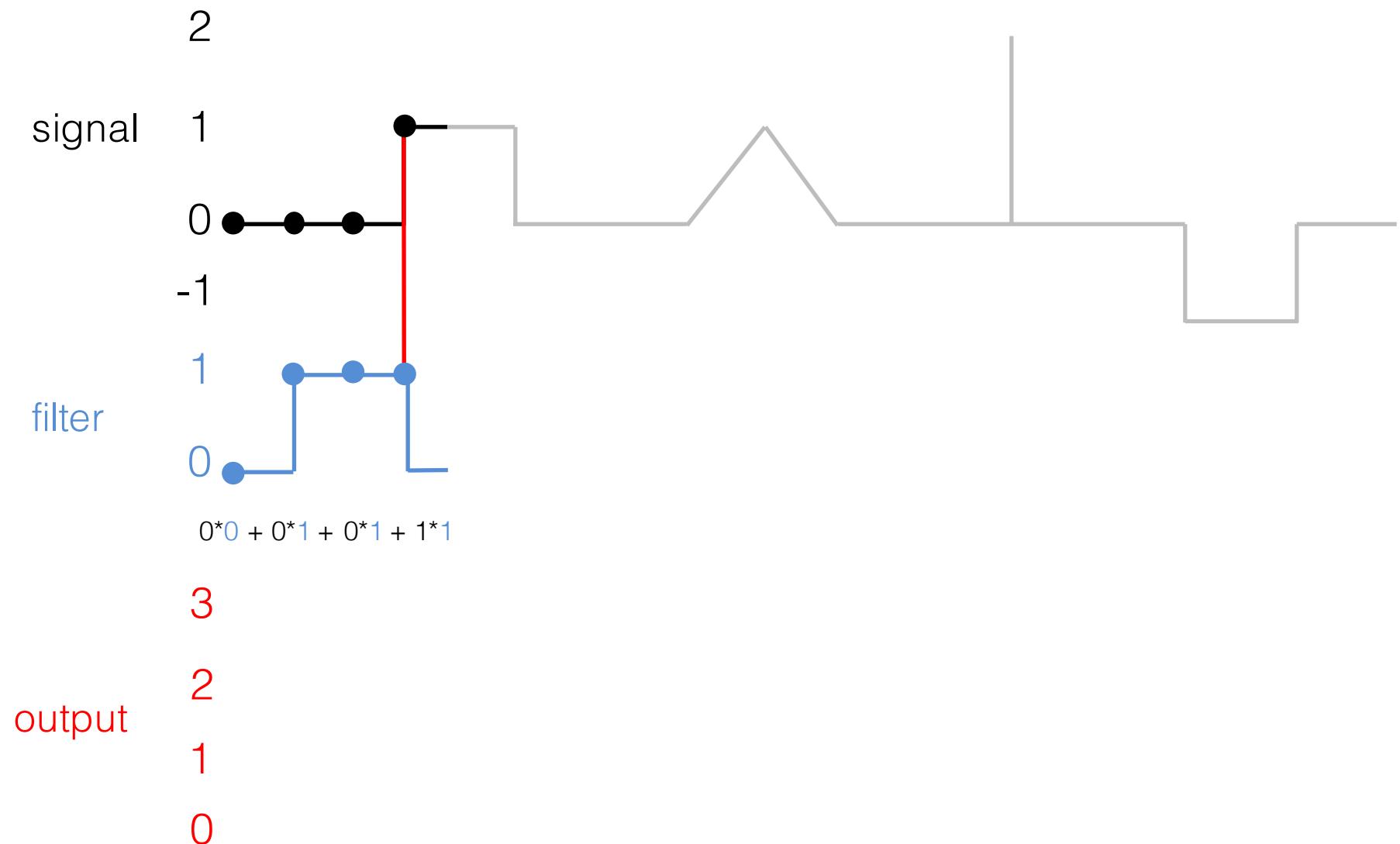


MATLAB: `conv(signal, filter, 'valid')`

Convolution

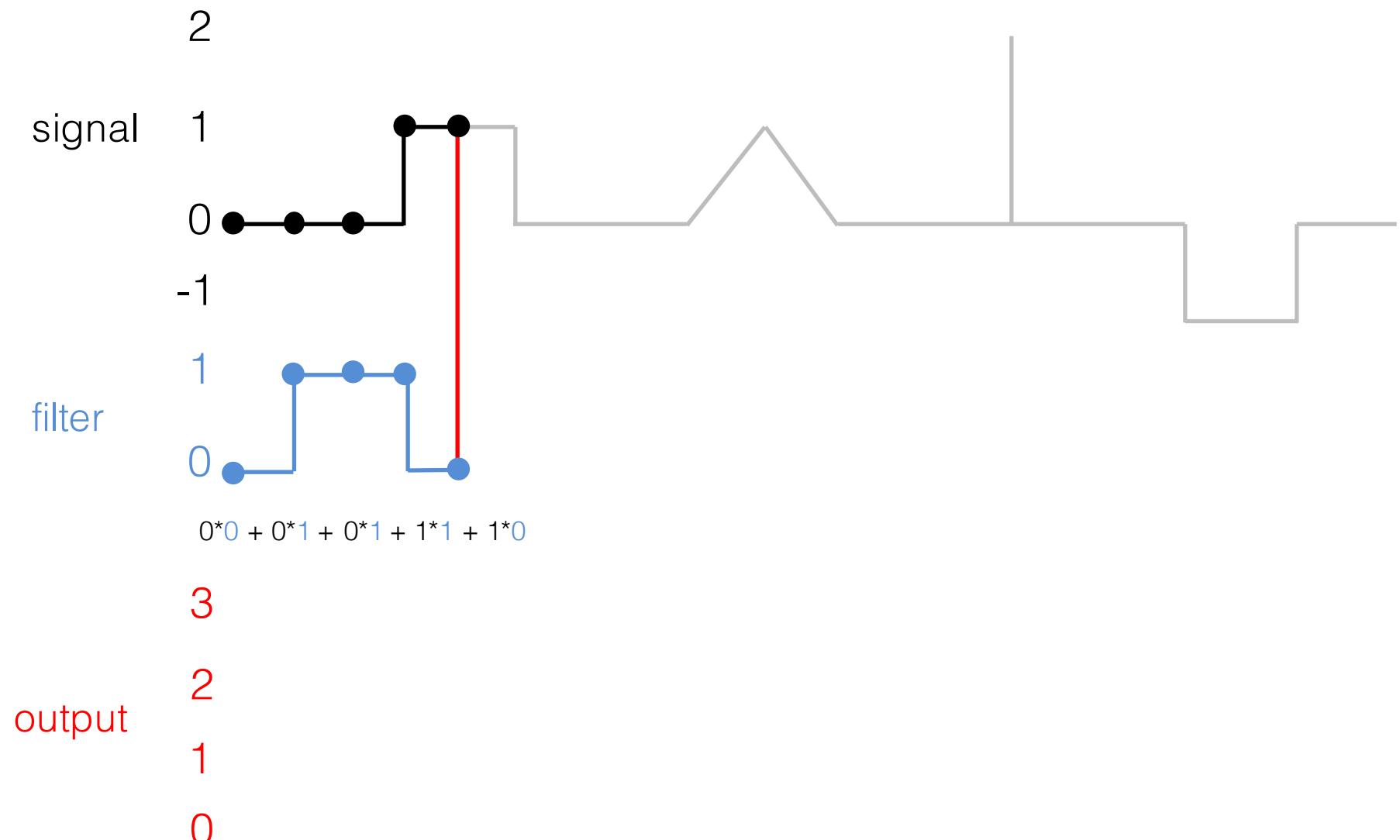


Convolution

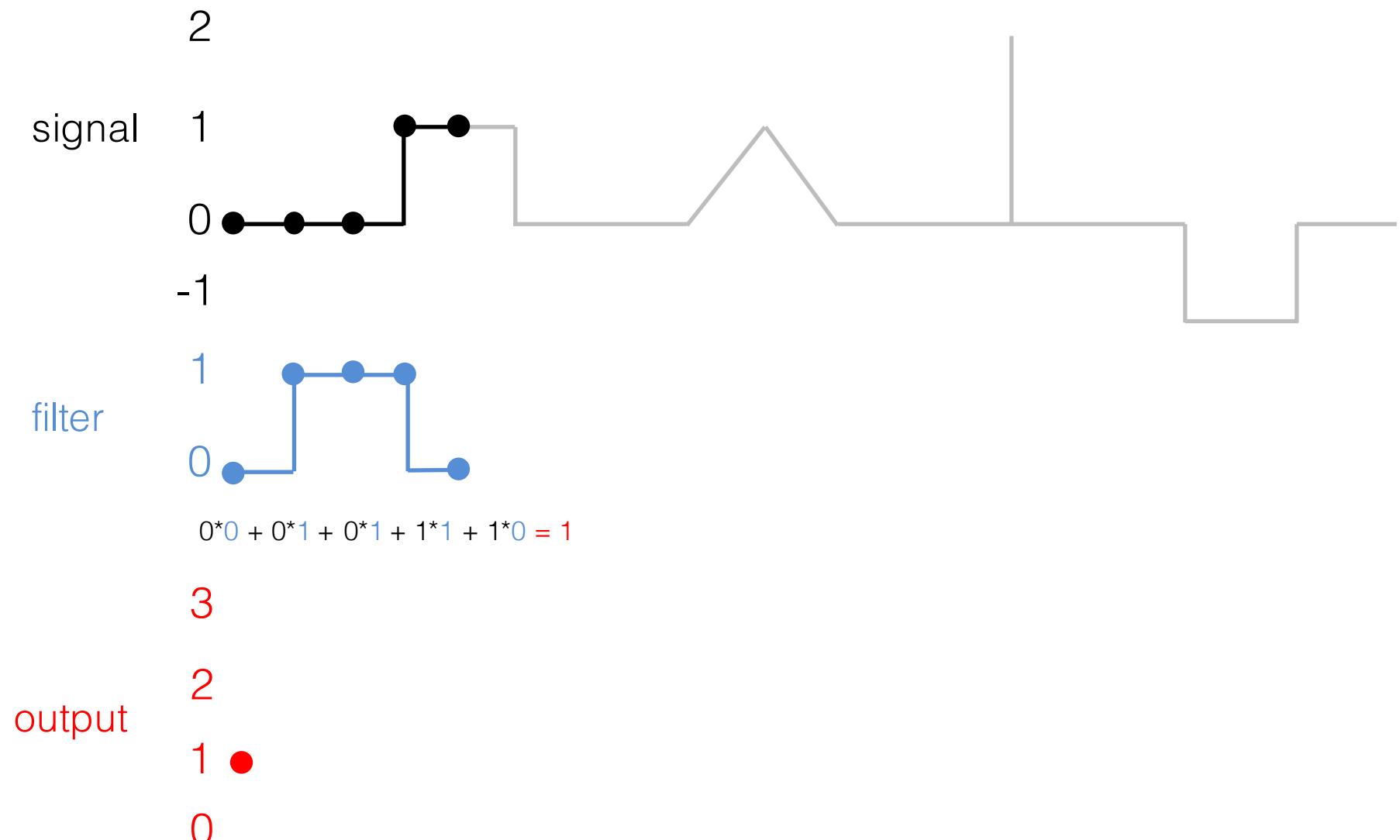


MATLAB: `conv(signal, filter, 'valid')`

Convolution

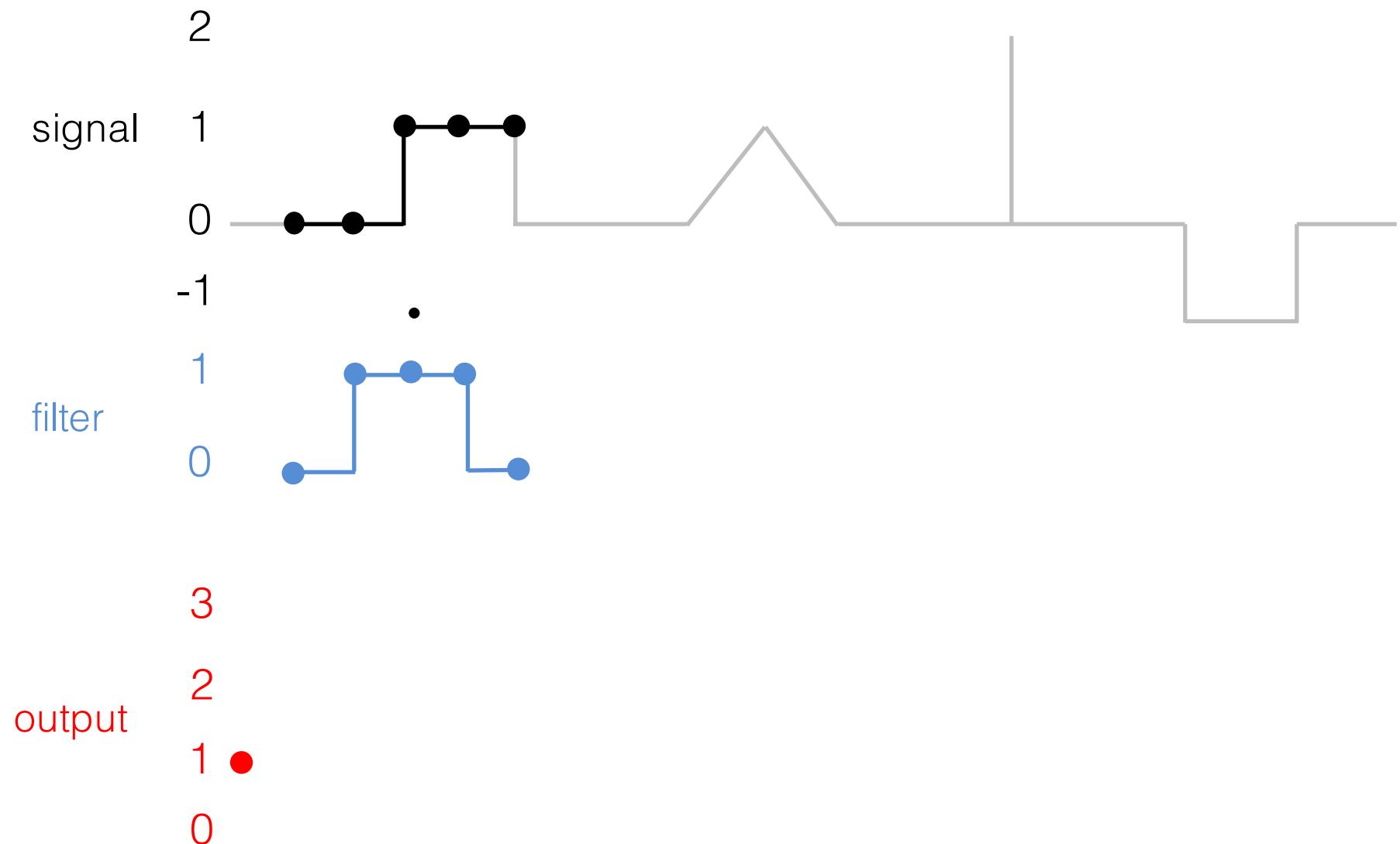


Convolution



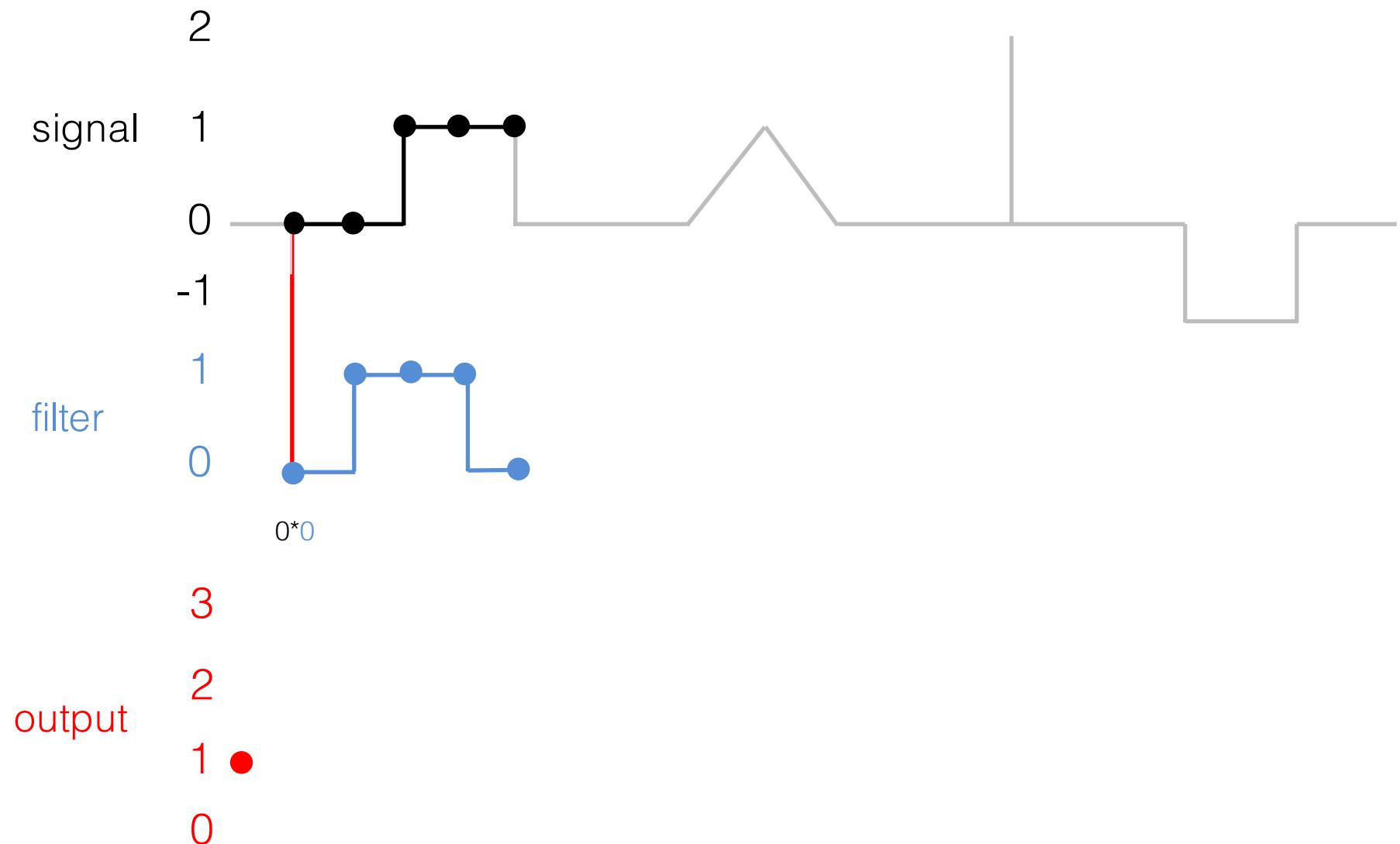
MATLAB: `conv(signal, filter, 'valid')`

Convolution



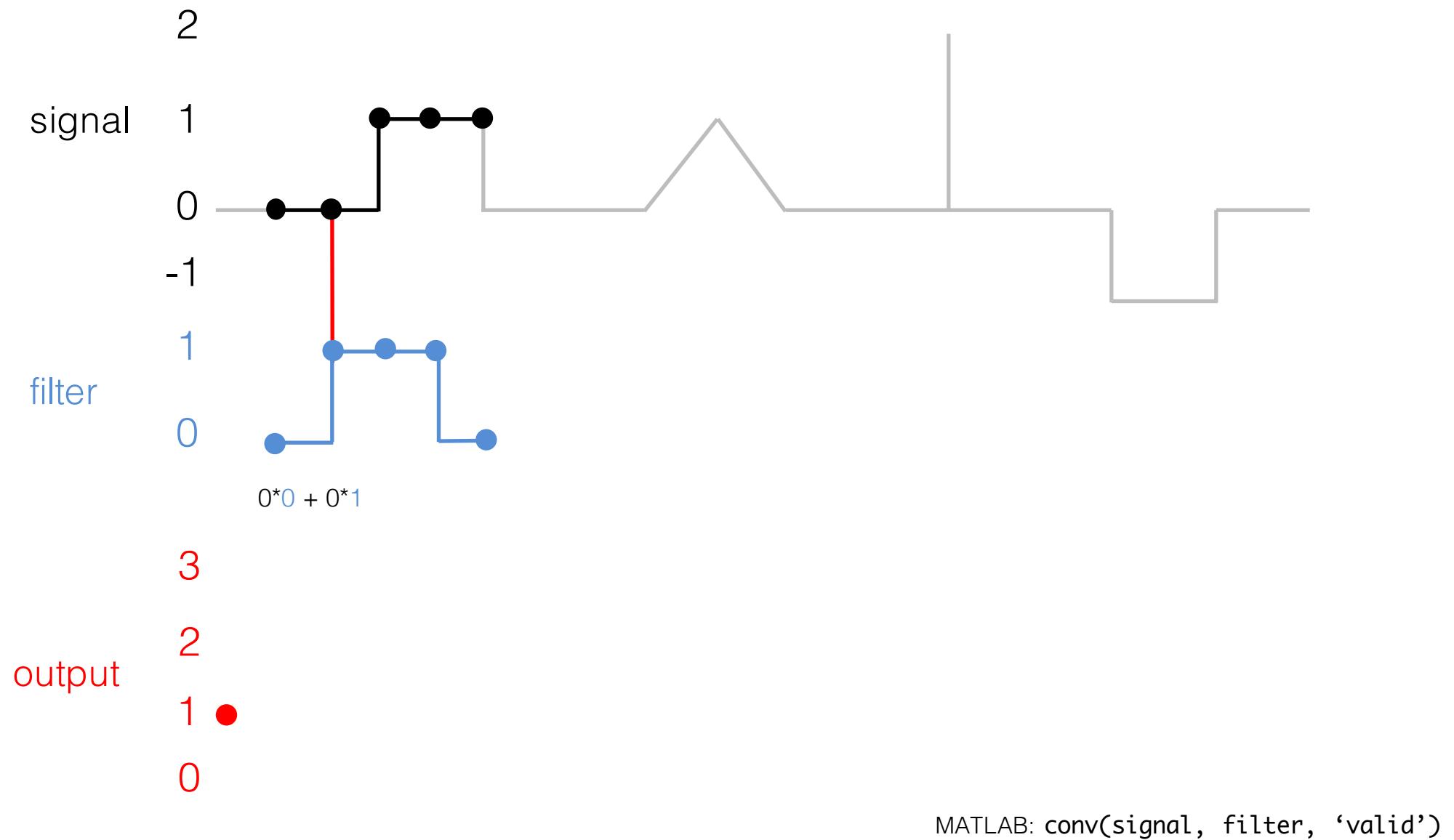
MATLAB: `conv(signal, filter, 'valid')`

Convolution

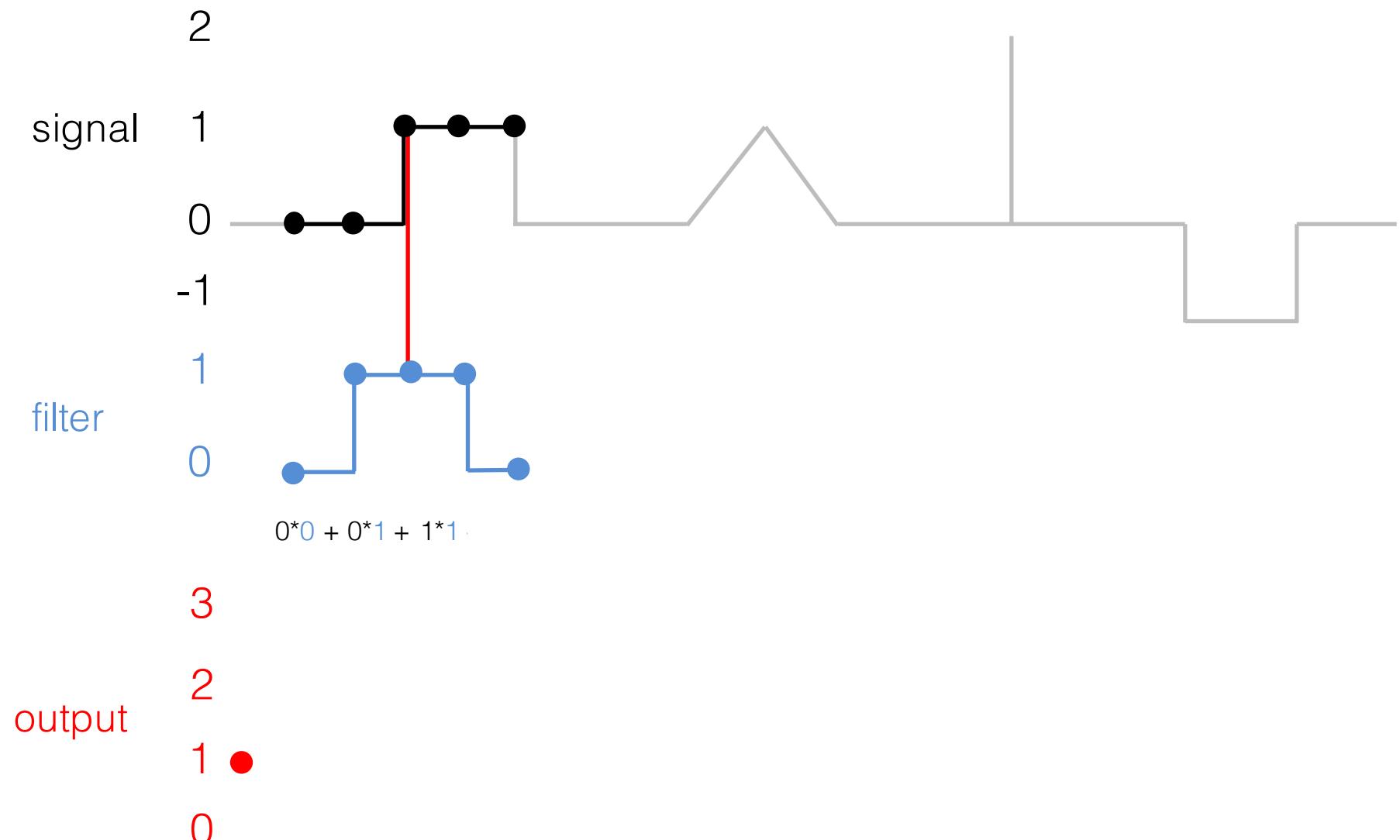


MATLAB: `conv(signal, filter, 'valid')`

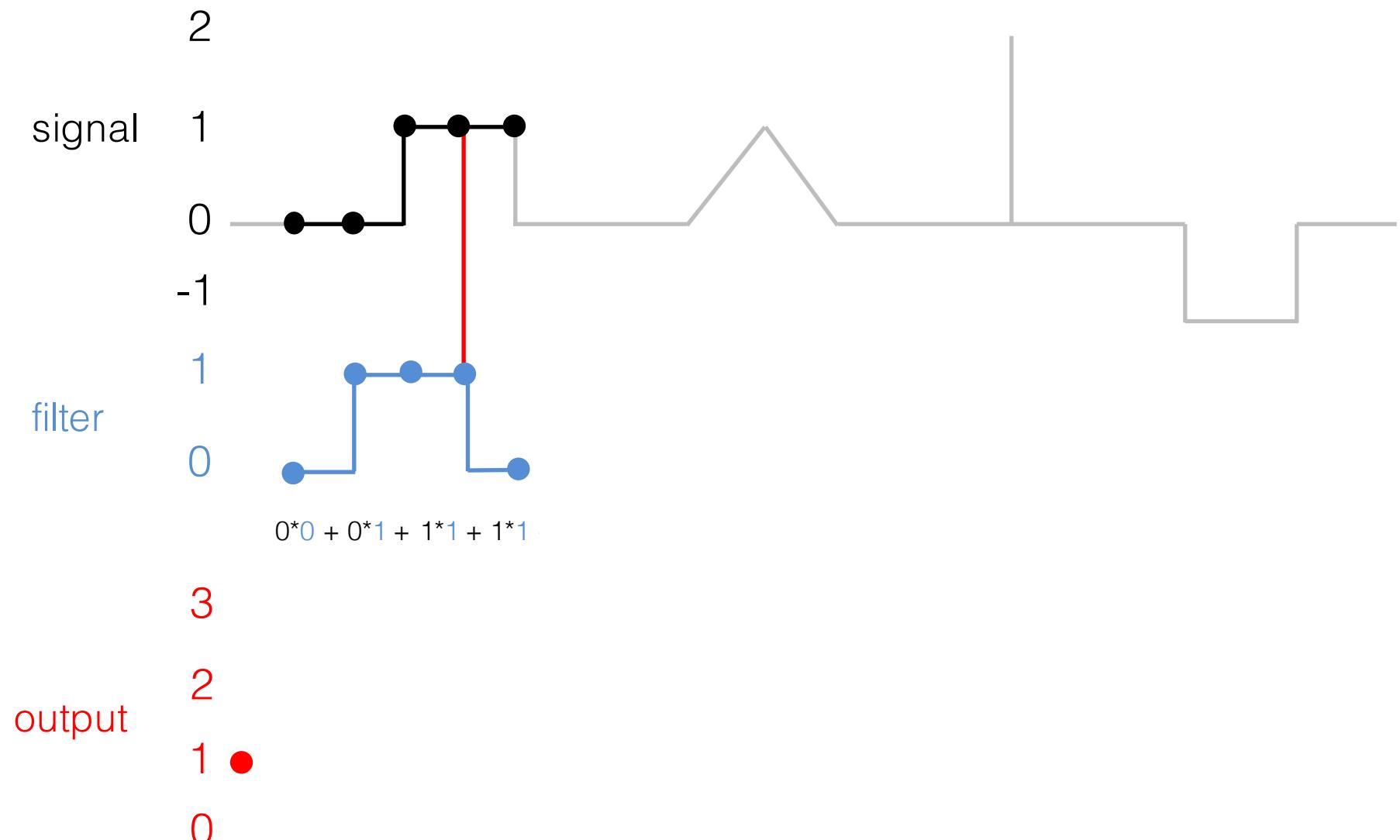
Convolution



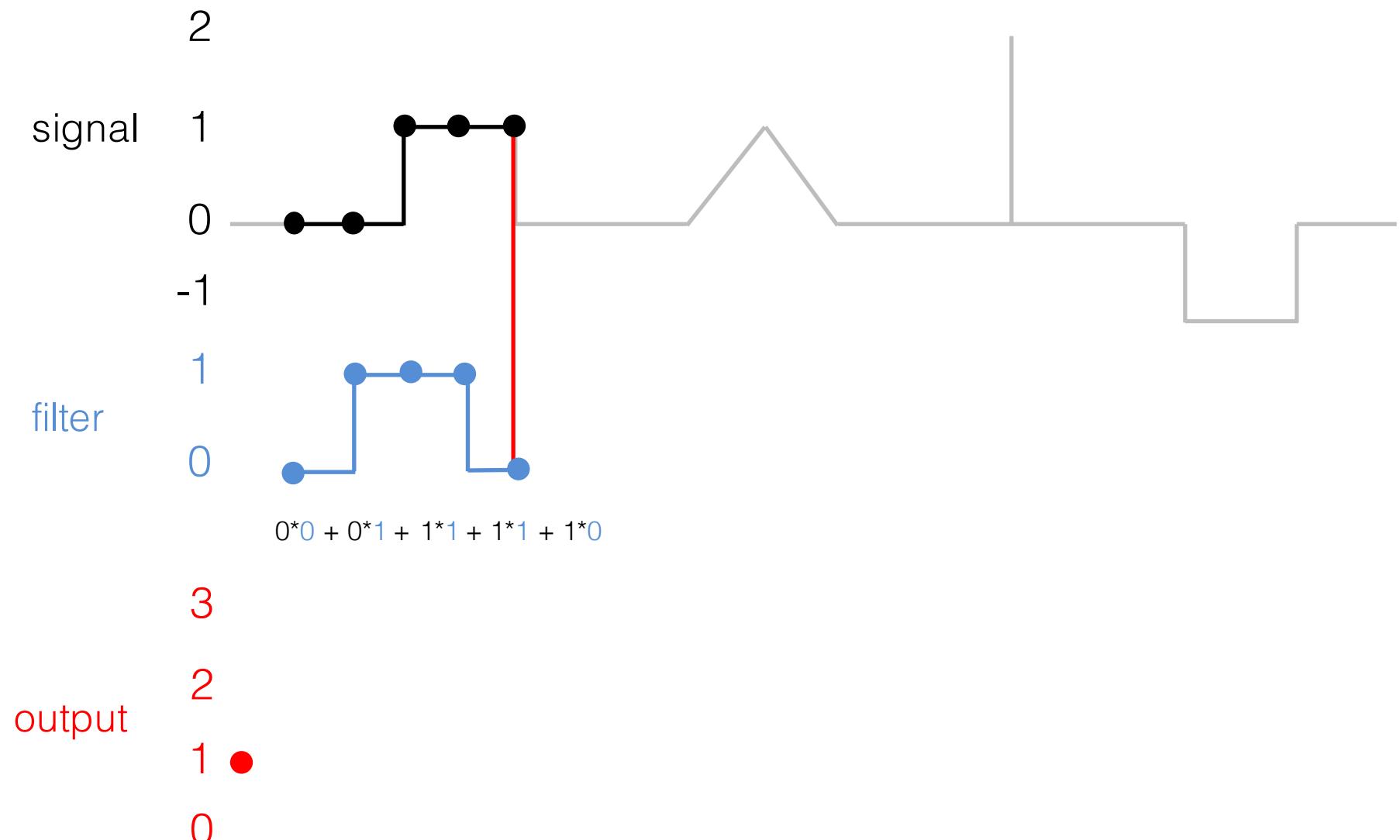
Convolution



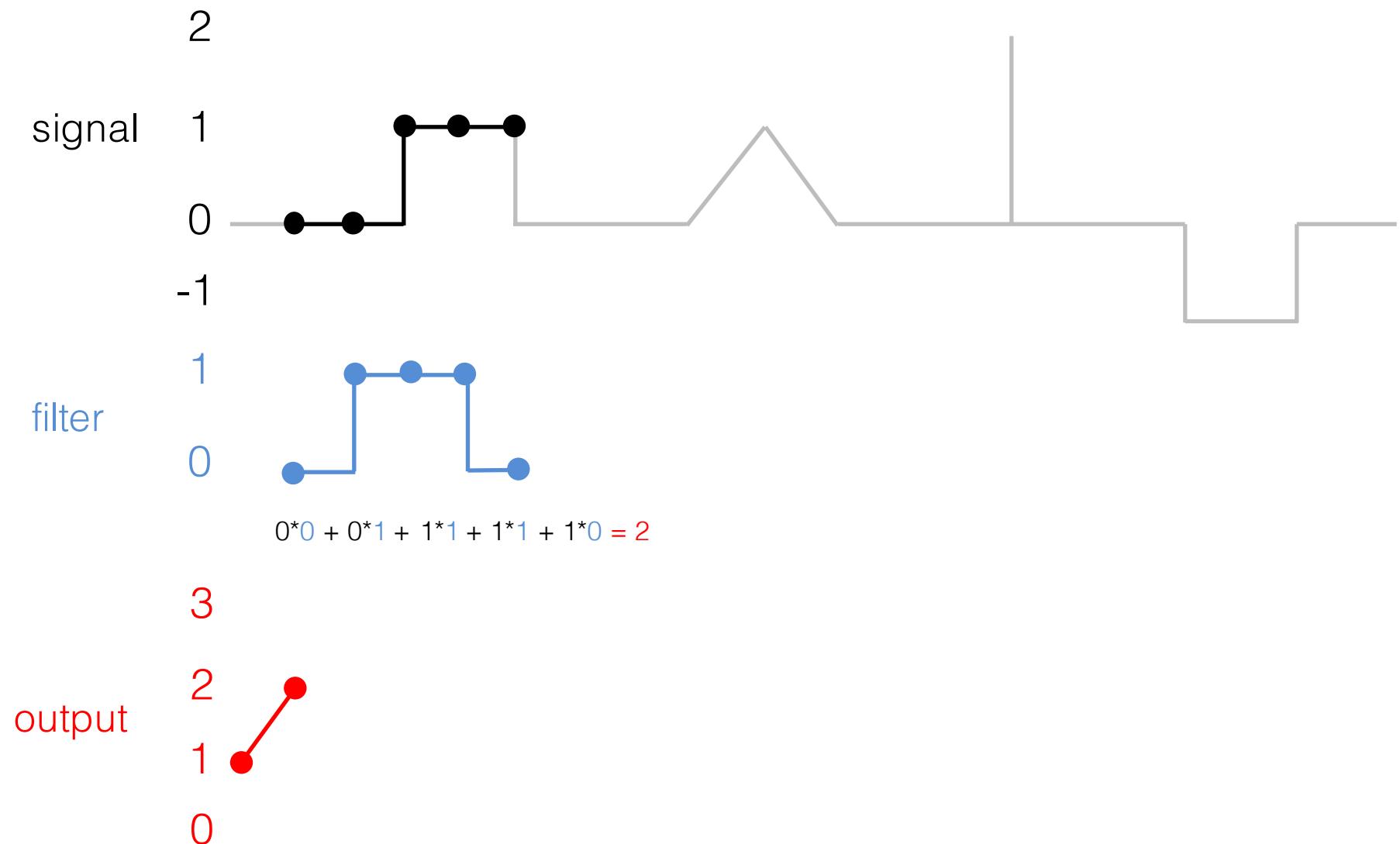
Convolution



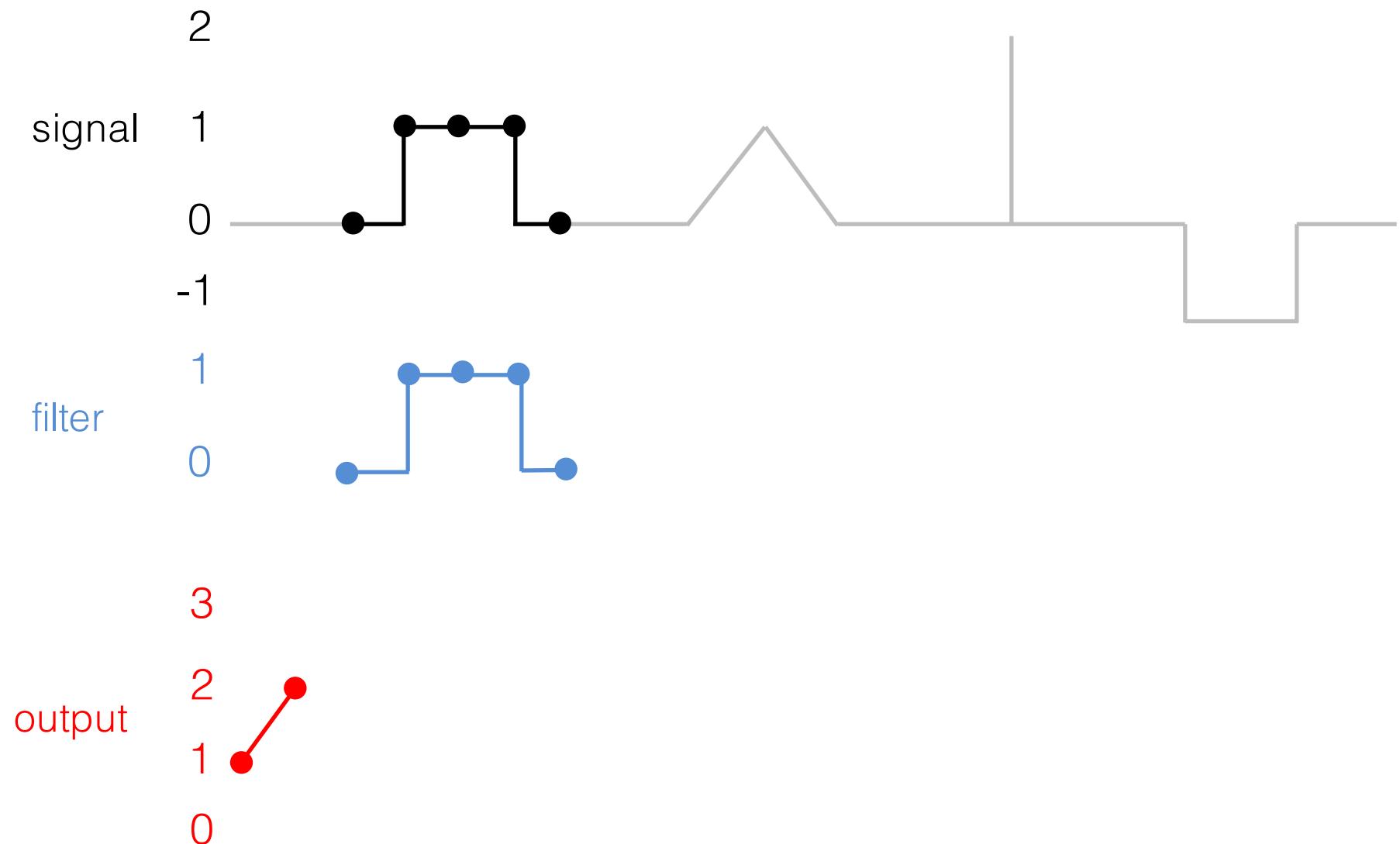
Convolution



Convolution

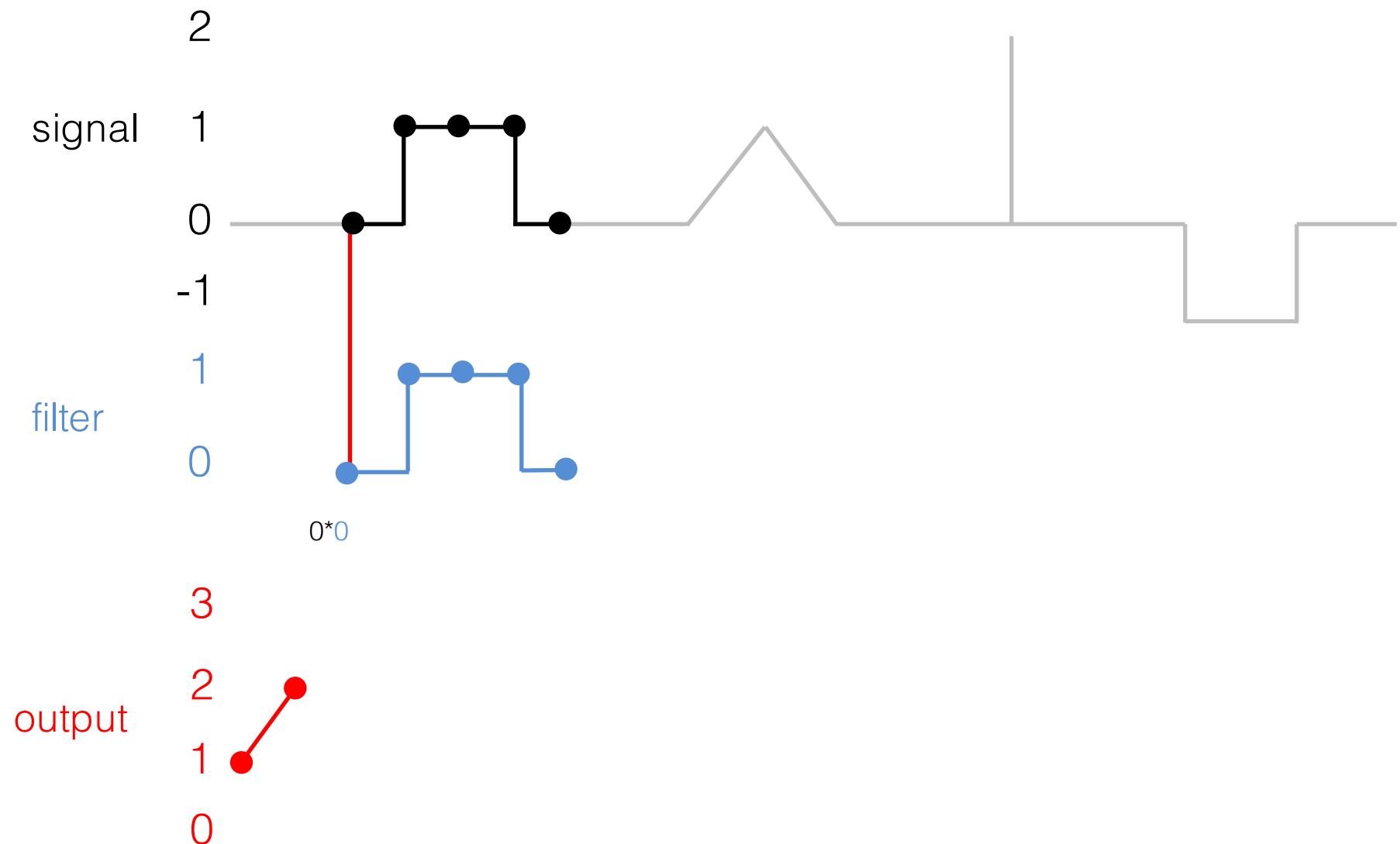


Convolution

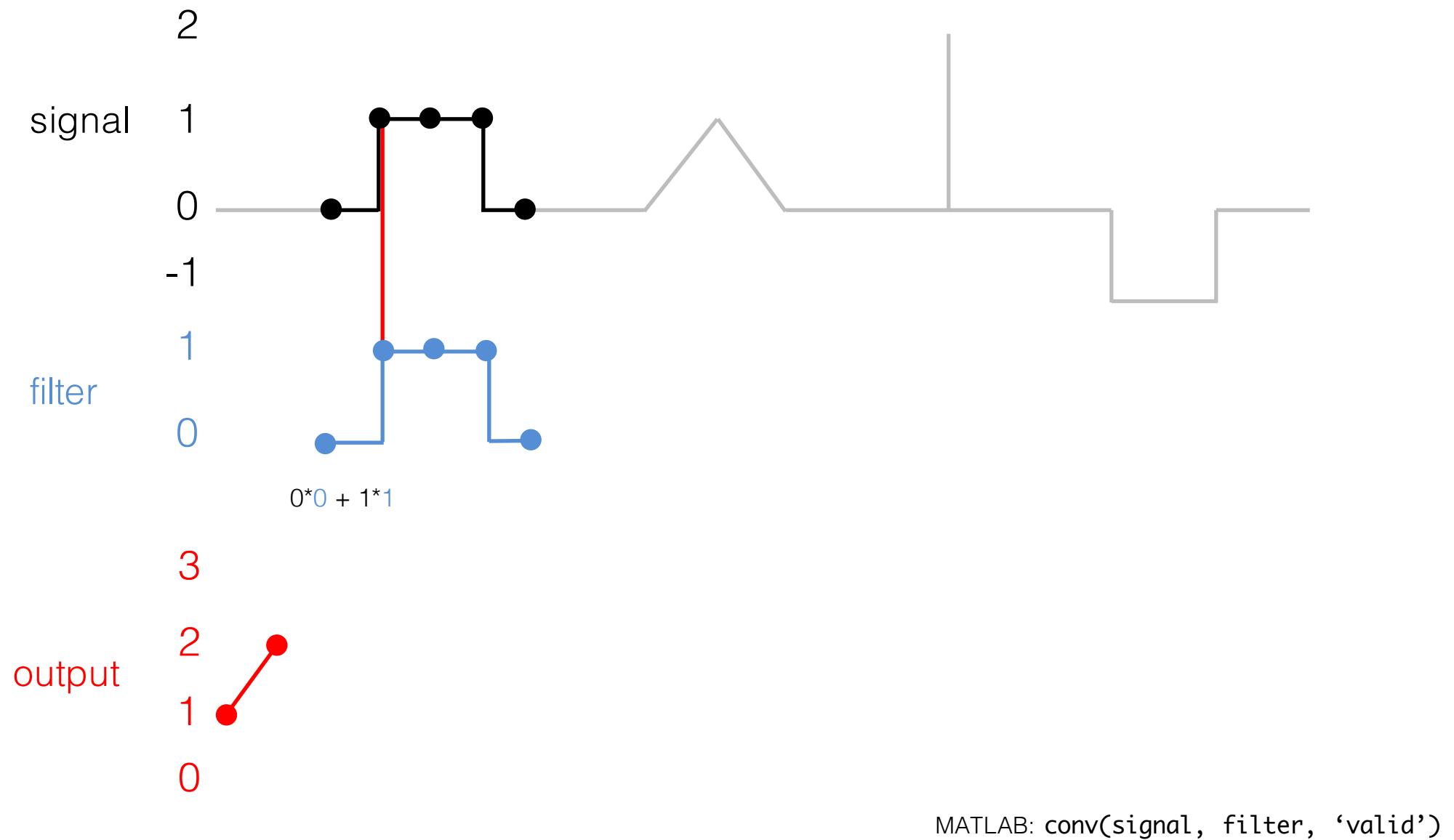


MATLAB: `conv(signal, filter, 'valid')`

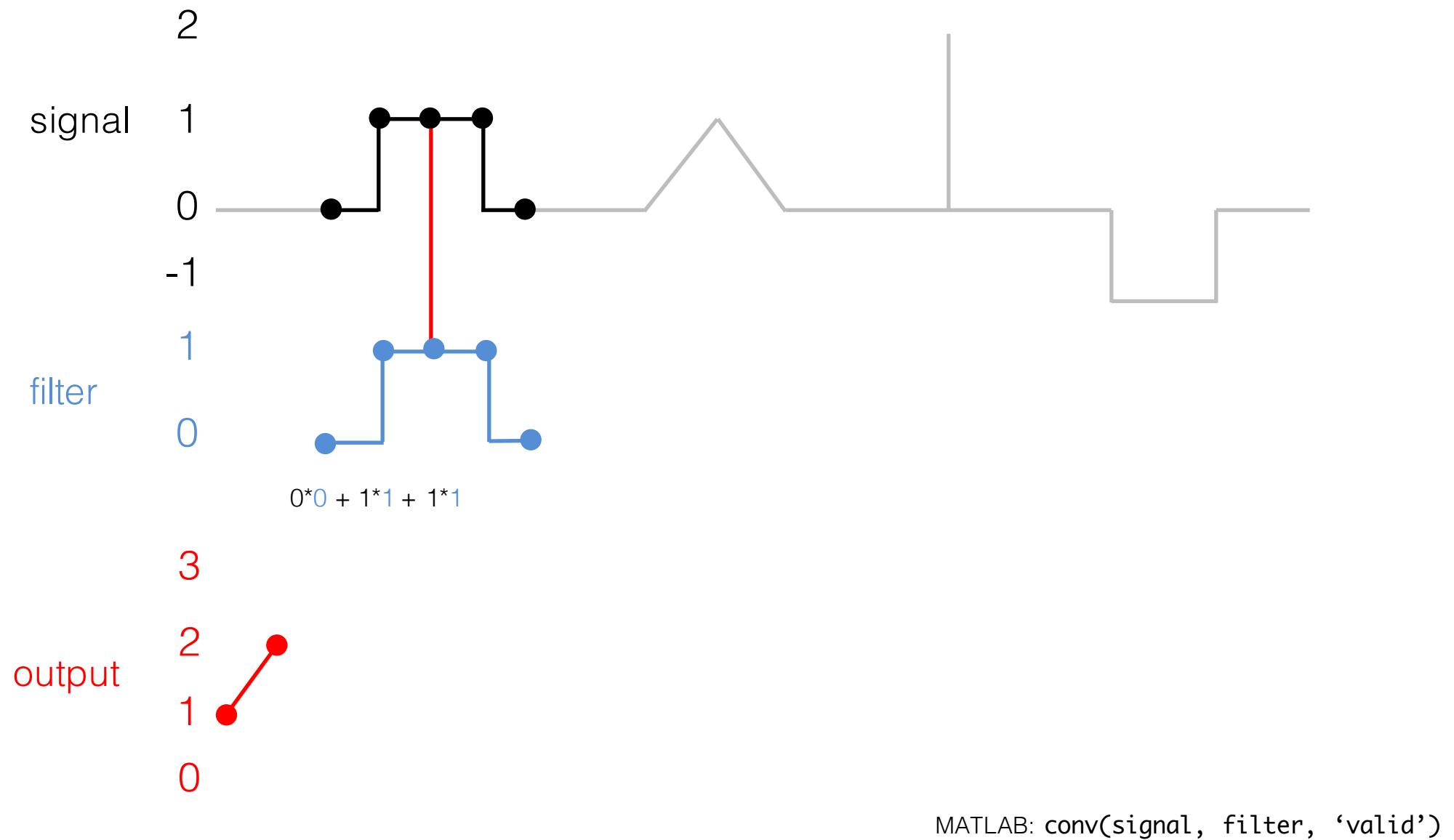
Convolution



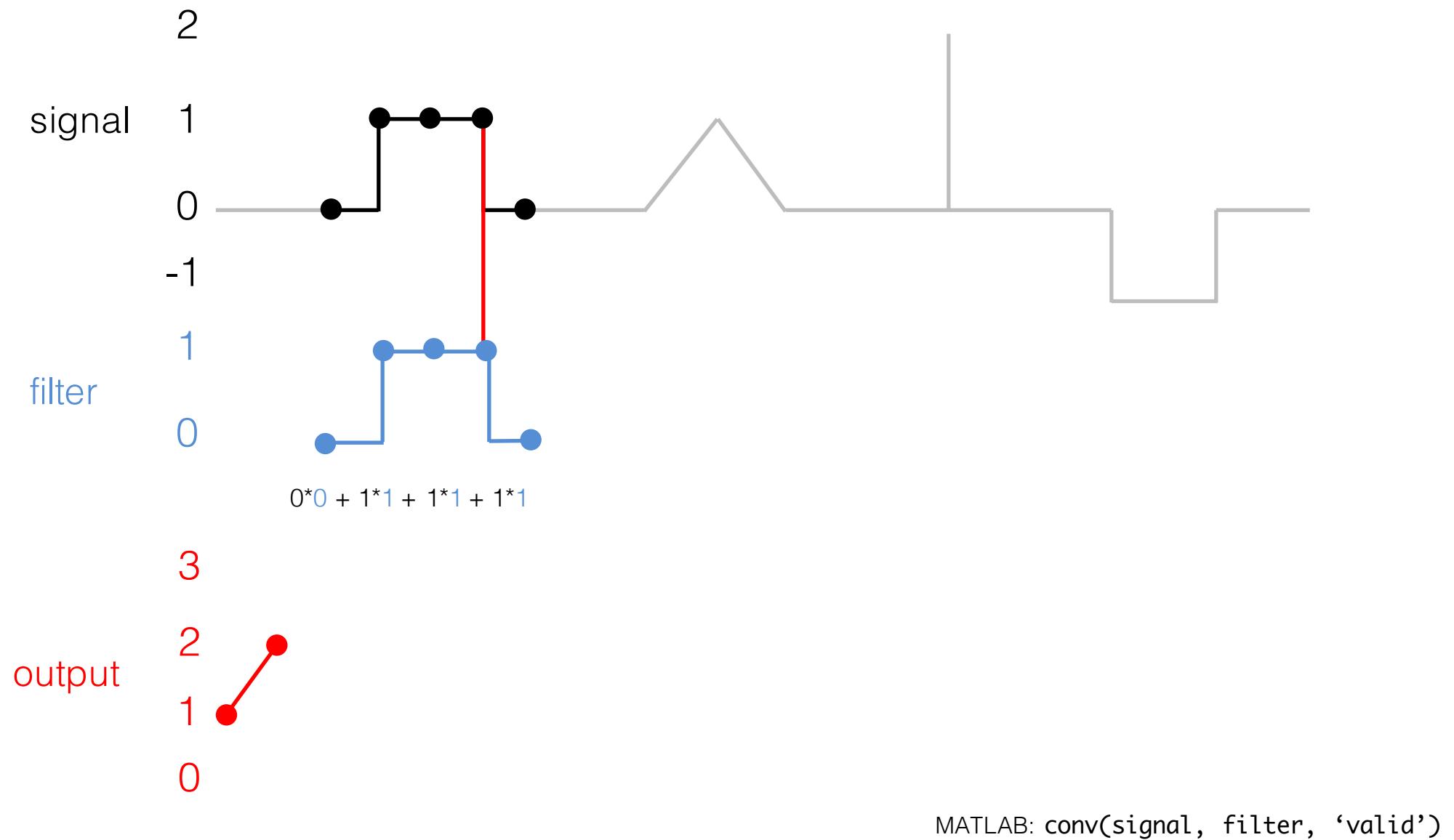
Convolution



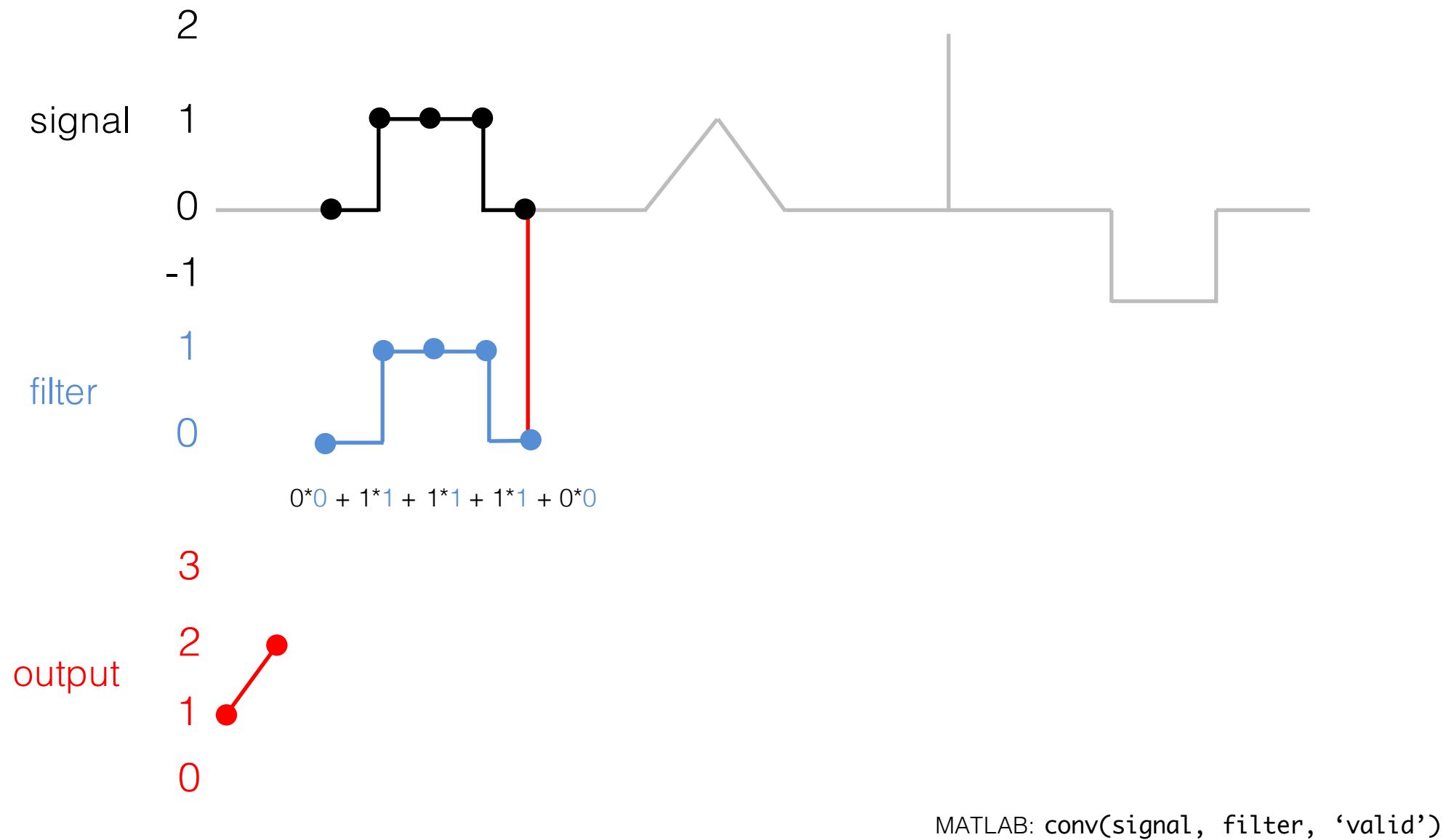
Convolution



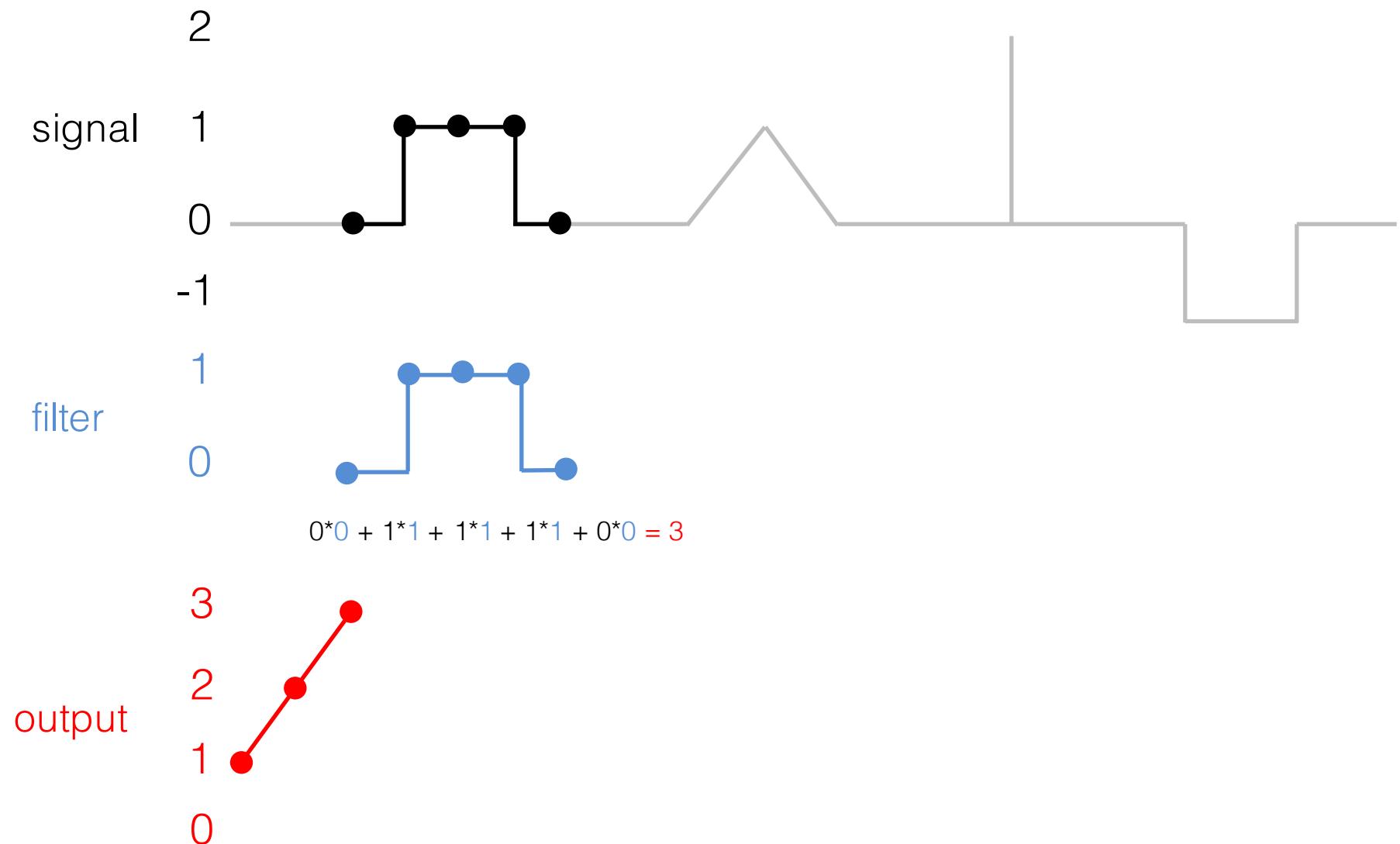
Convolution



Convolution

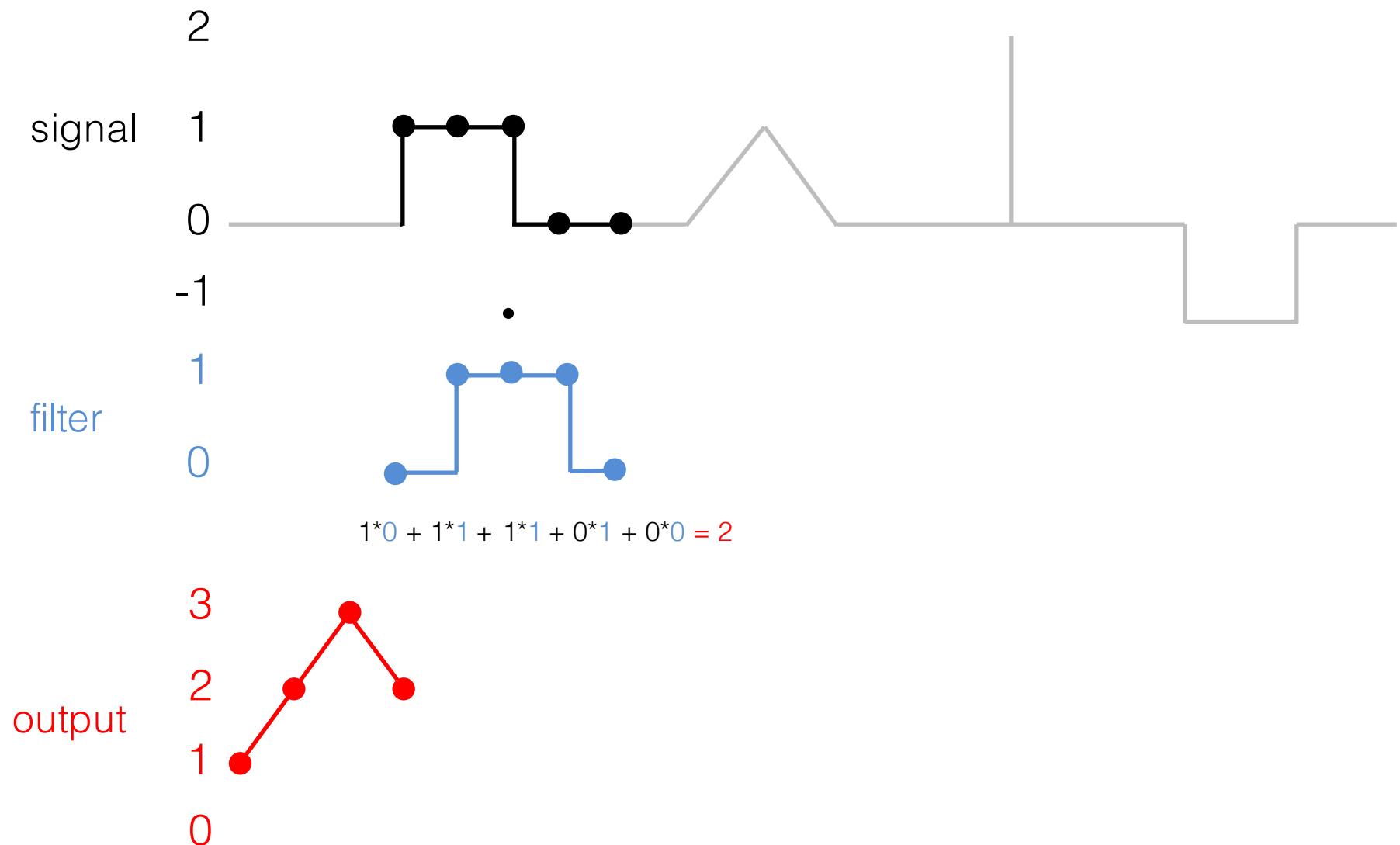


Convolution



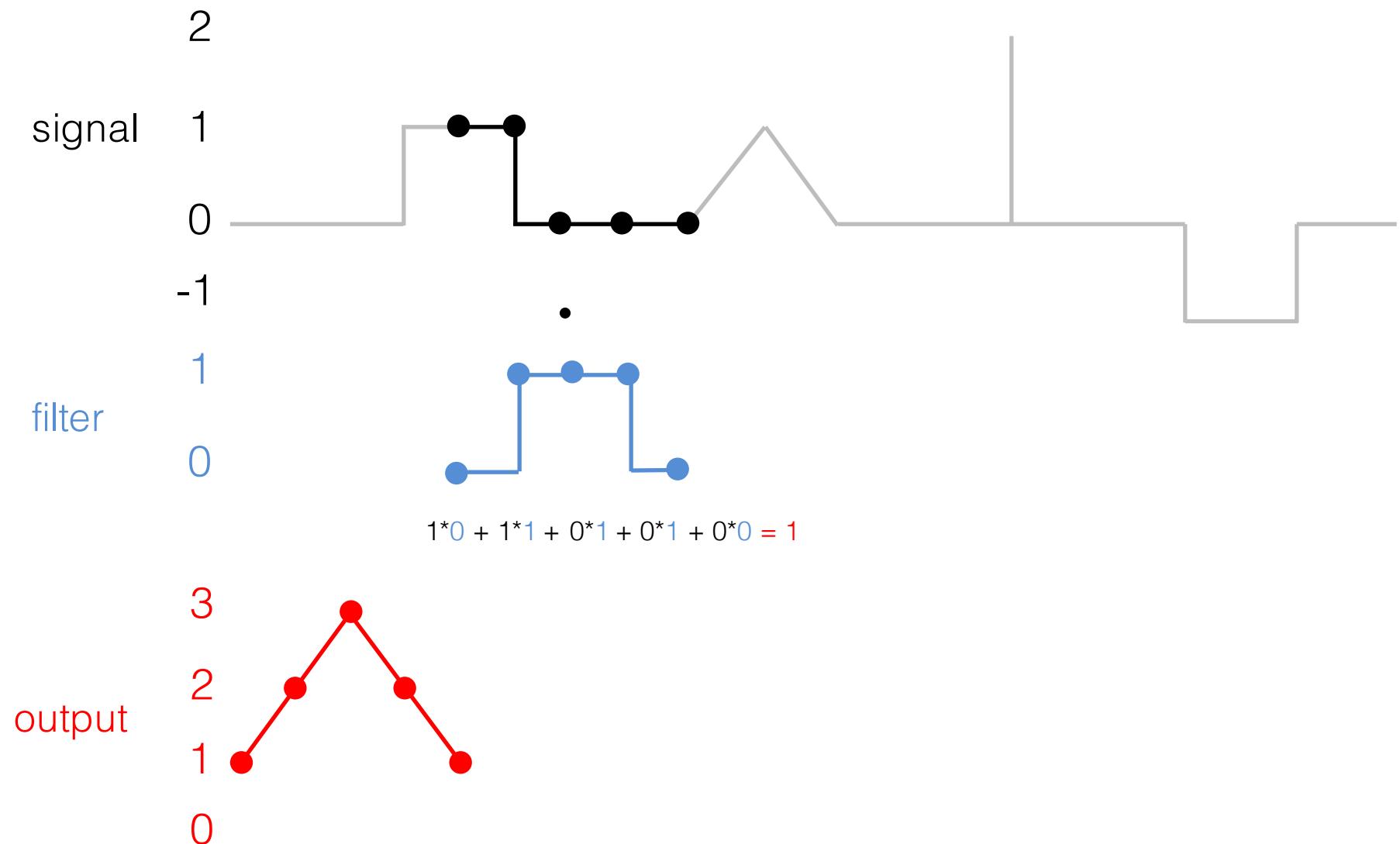
MATLAB: `conv(signal, filter, 'valid')`

Convolution



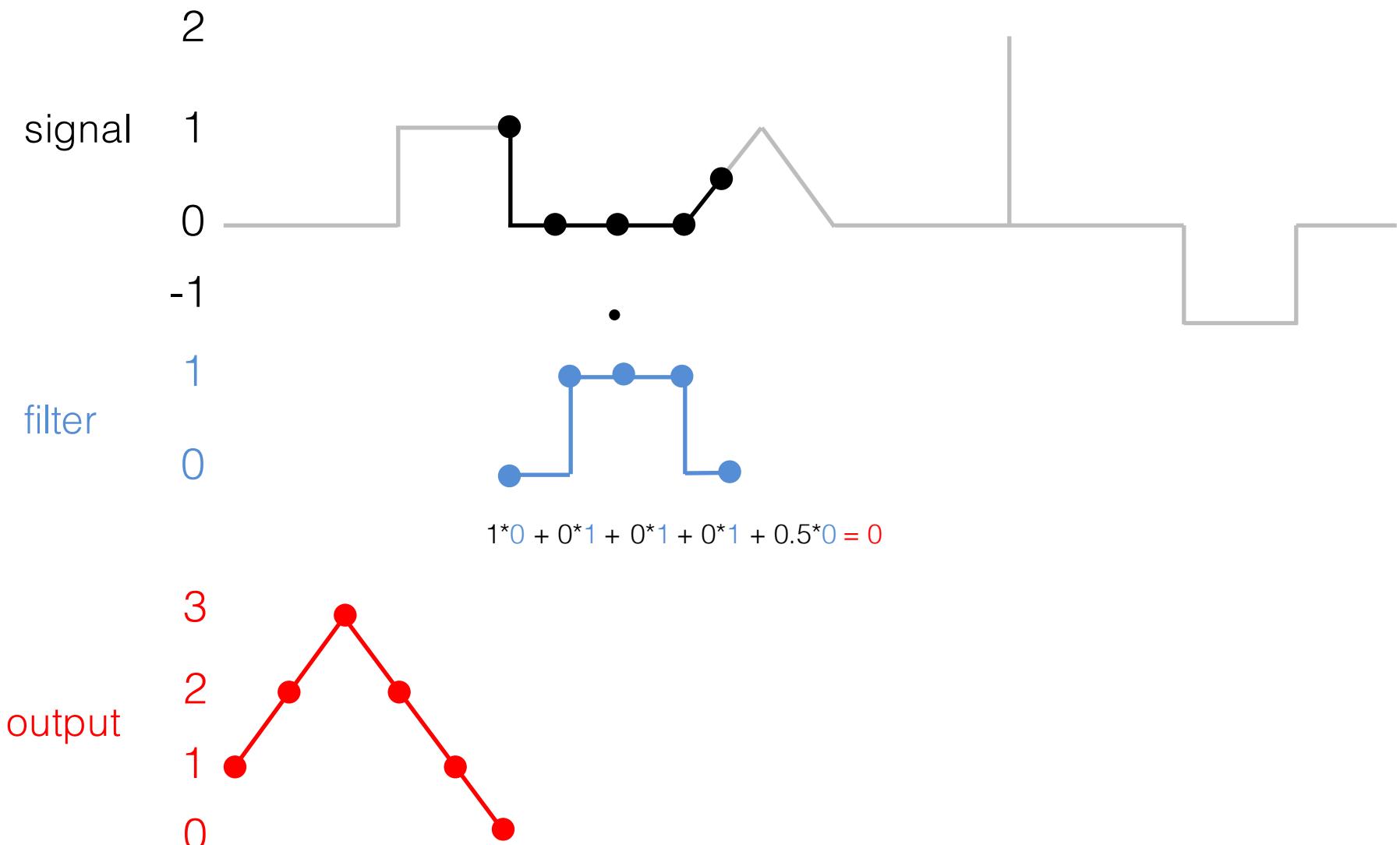
MATLAB: `conv(signal, filter, 'valid')`

Convolution



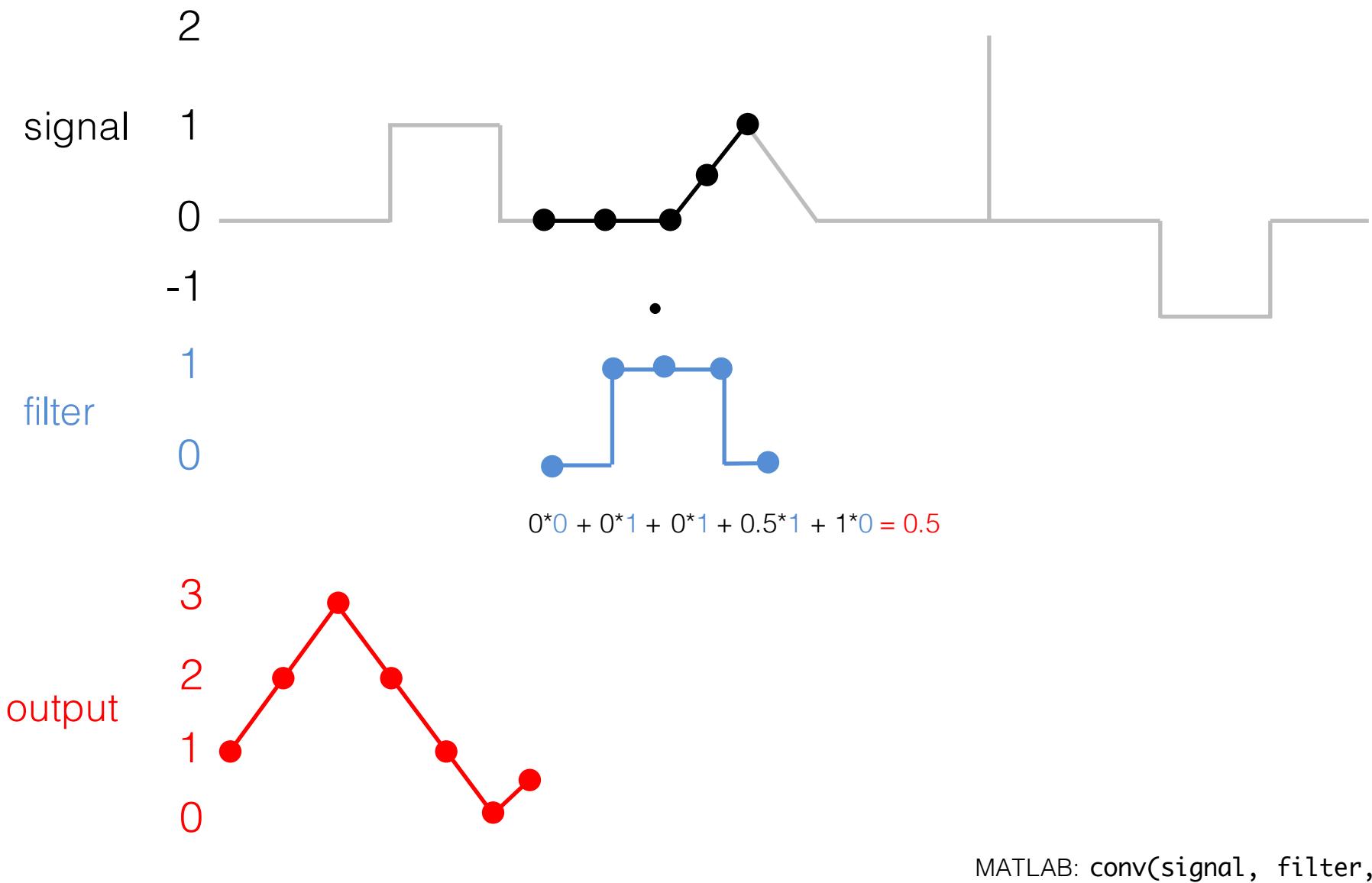
MATLAB: `conv(signal, filter, 'valid')`

Convolution

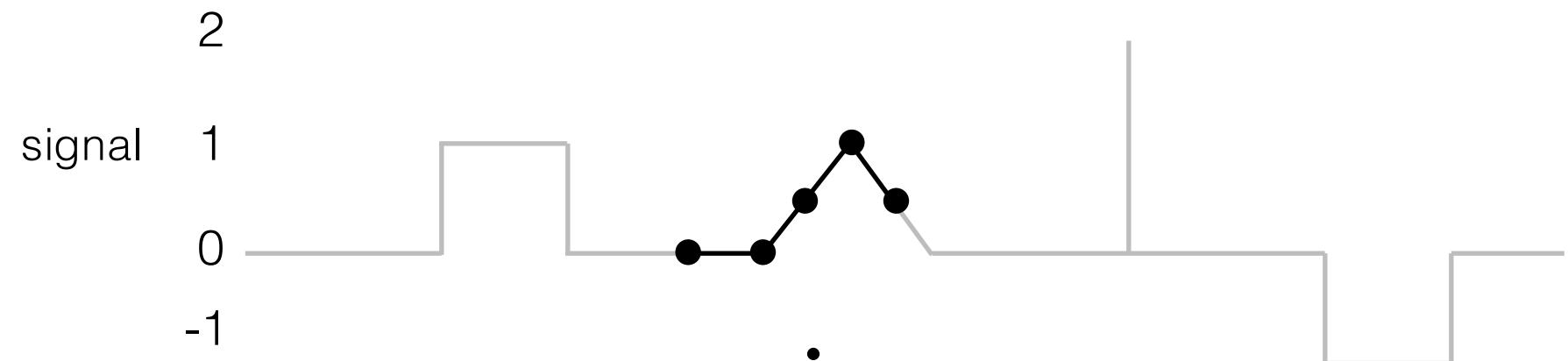


MATLAB: `conv(signal, filter, 'valid')`

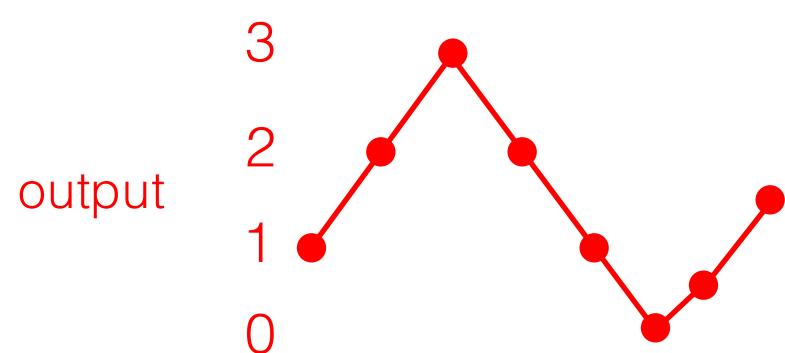
Convolution



Convolution

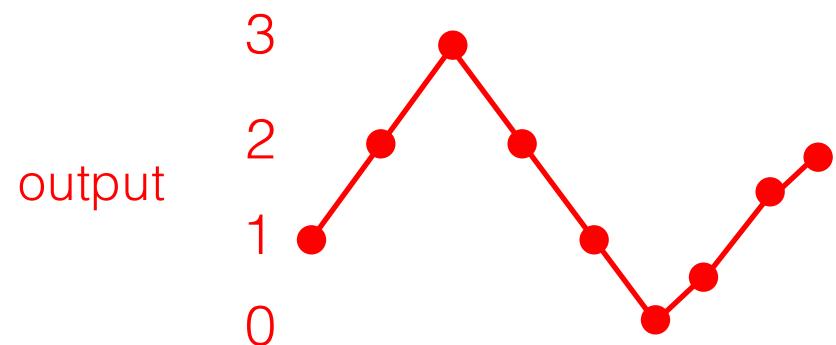
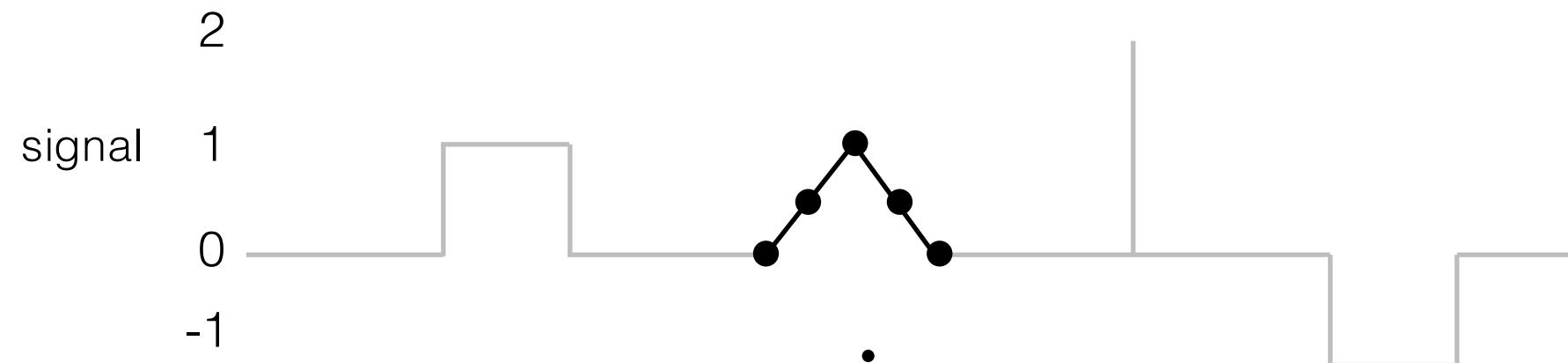


$$0*0 + 0*1 + 0*1 + 0.5*1 + 1*0 = 1.5$$



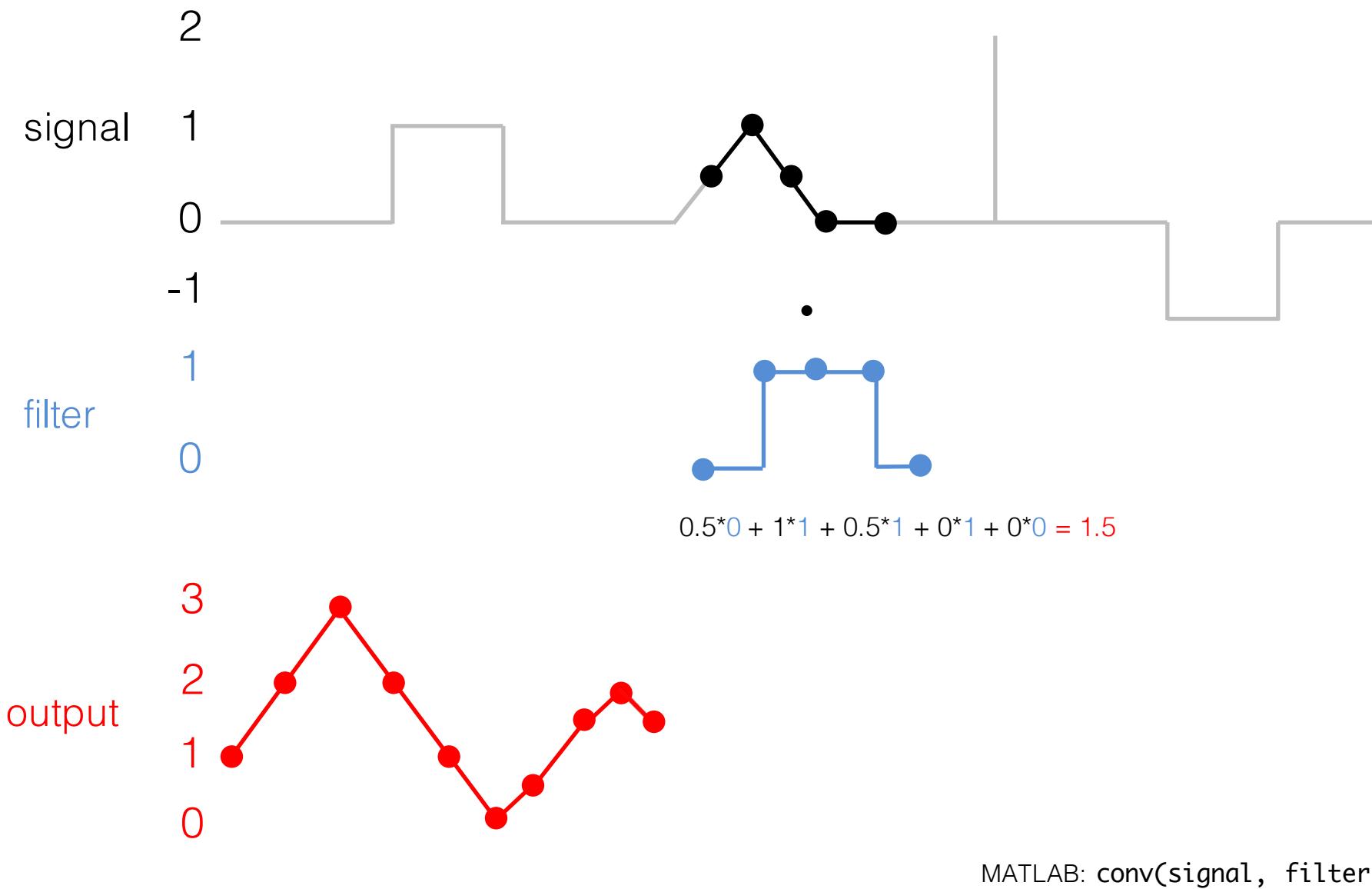
MATLAB: `conv(signal, filter, 'valid')`

Convolution

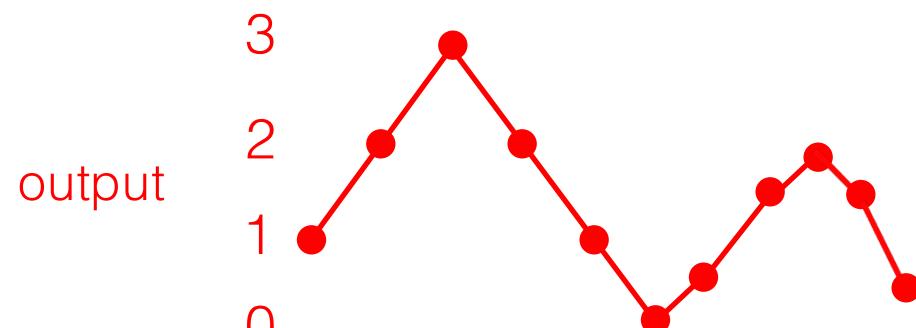
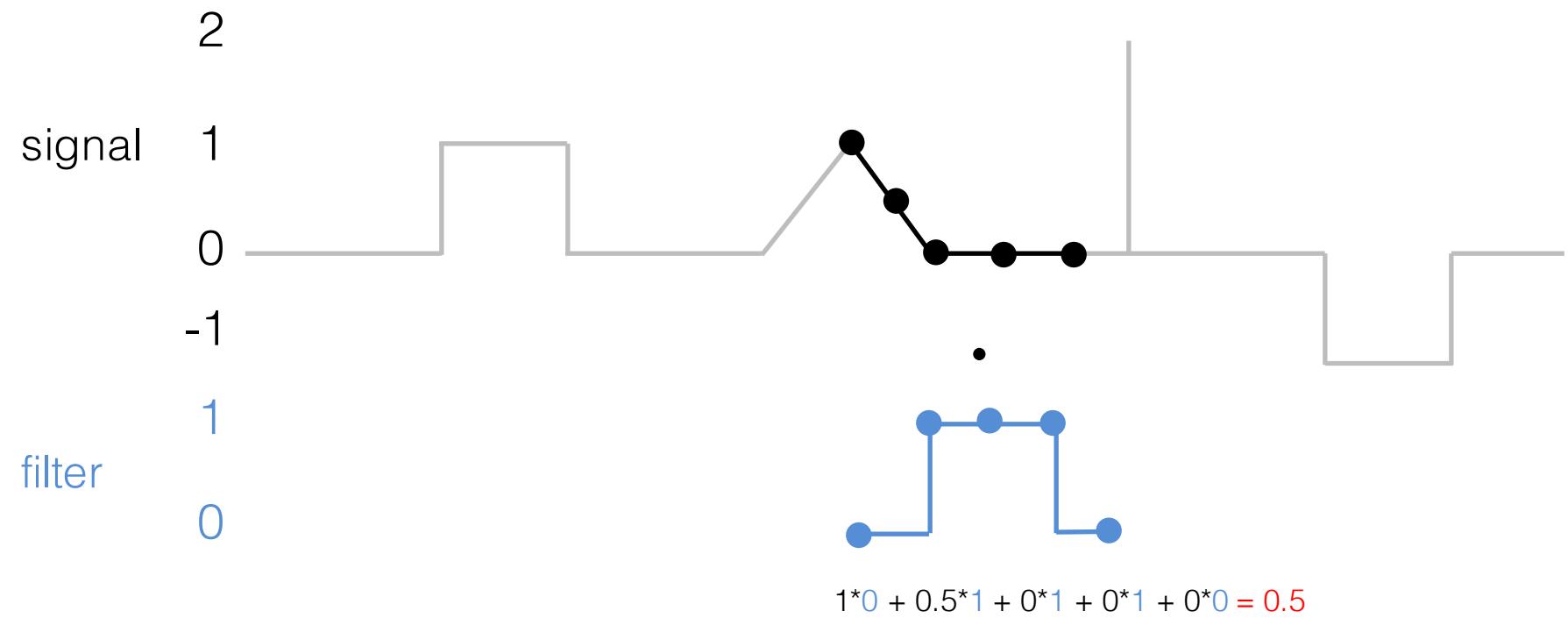


MATLAB: `conv(signal, filter, 'valid')`

Convolution

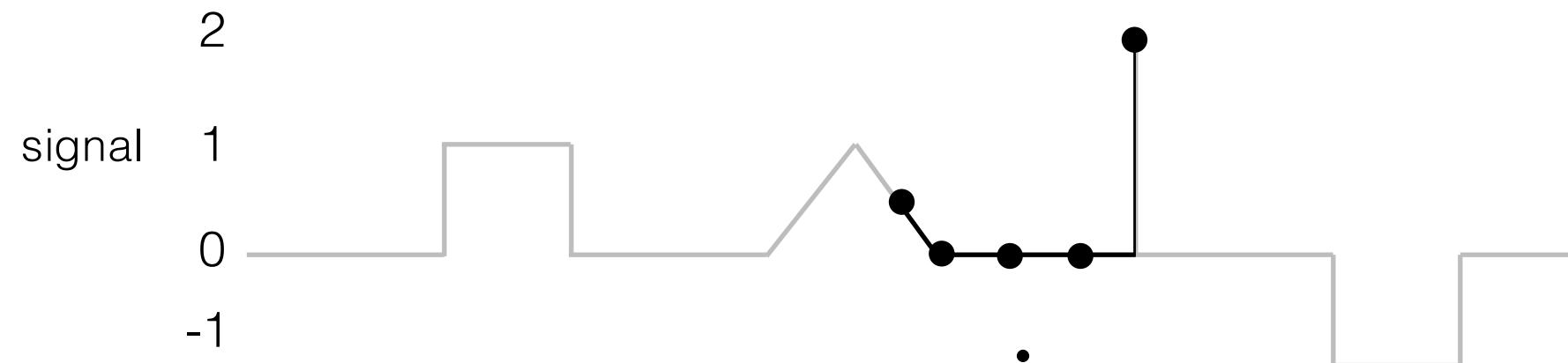


Convolution



MATLAB: `conv(signal, filter, 'valid')`

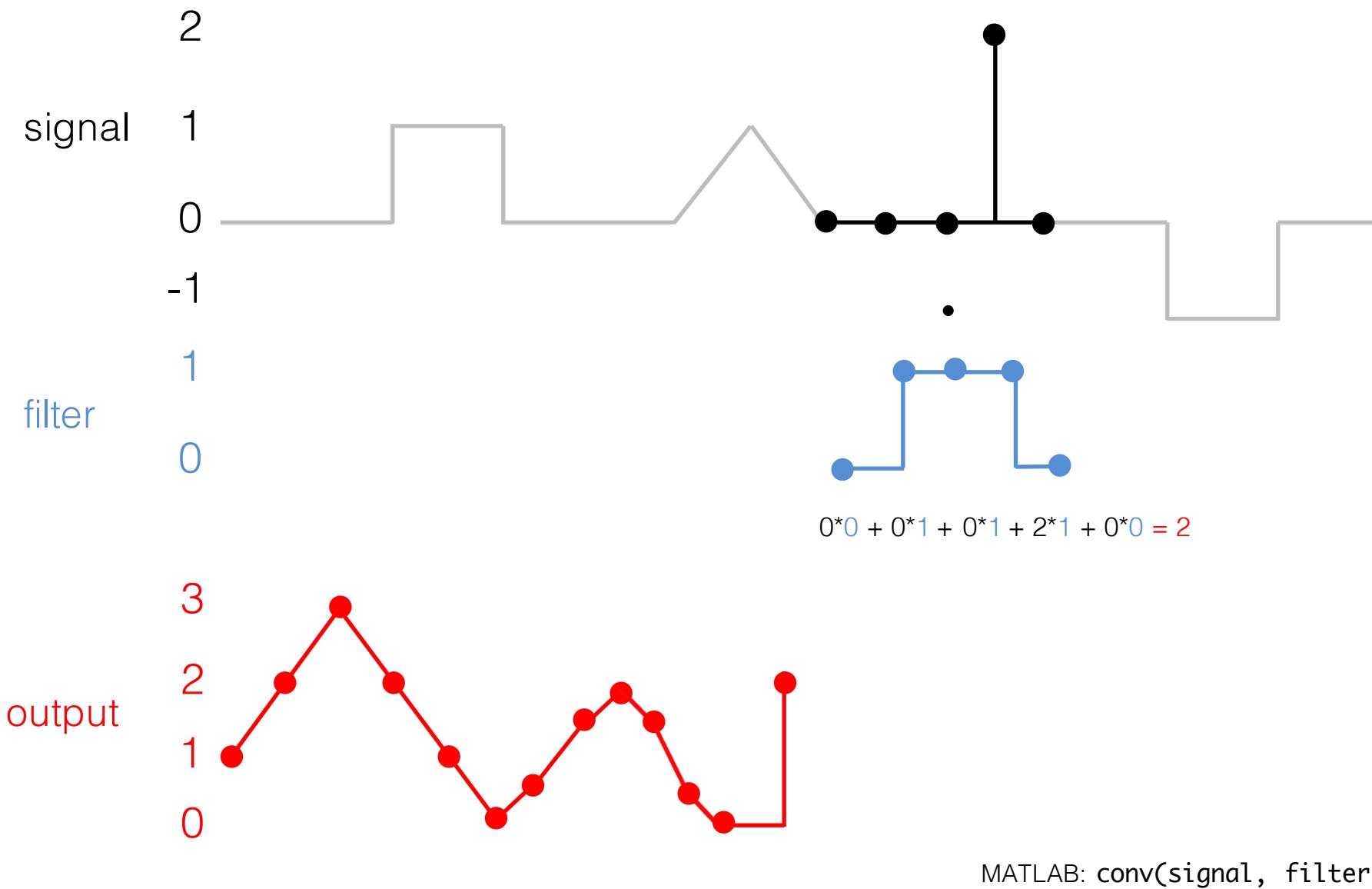
Convolution



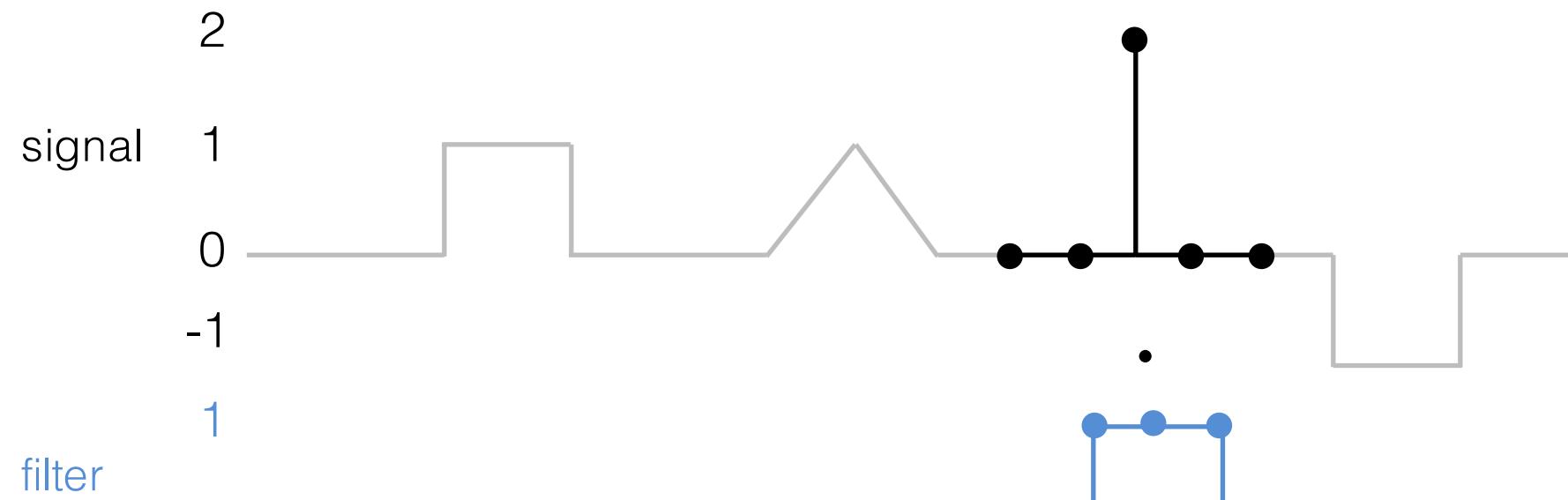
$$0.5*0 + 0*1 + 0*1 + 0*1 + 2*0 = 0$$

MATLAB: `conv(signal, filter, 'valid')`

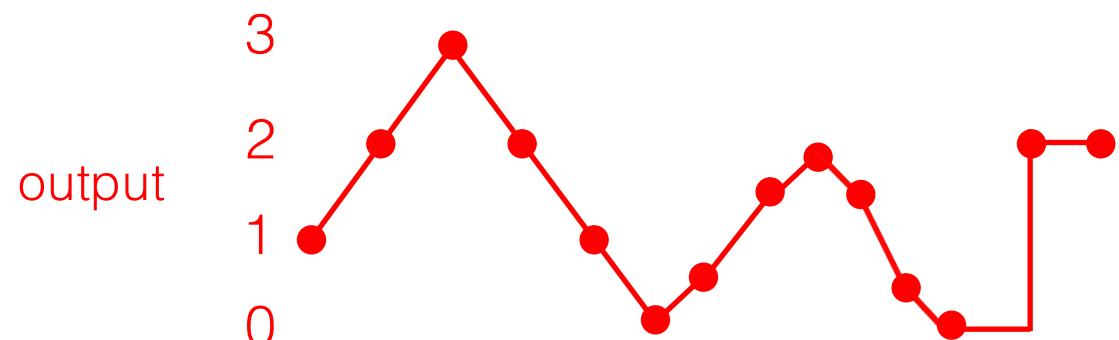
Convolution



Convolution

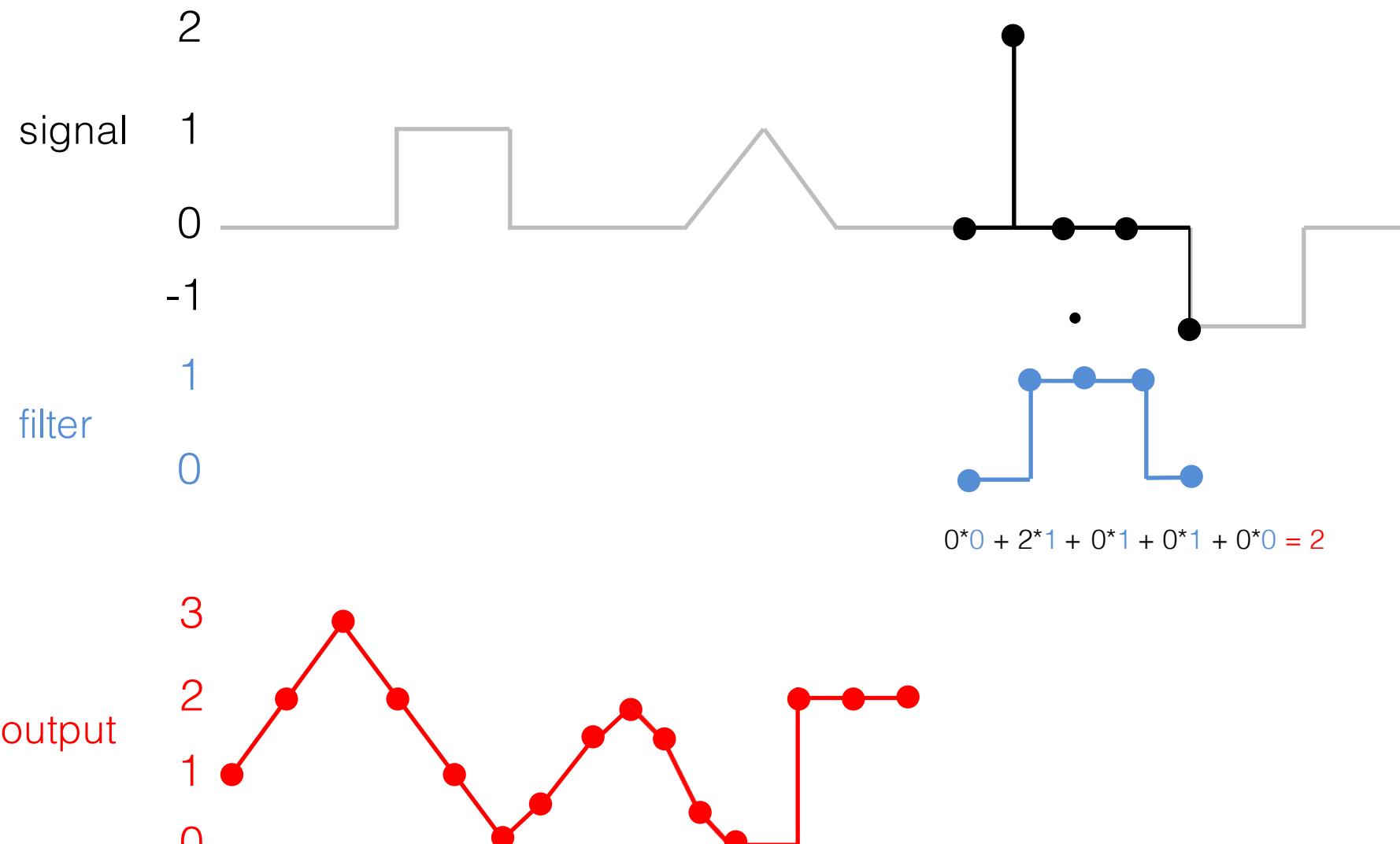


$$0*0 + 0*1 + 2*1 + 0*1 + 0*0 = 2$$



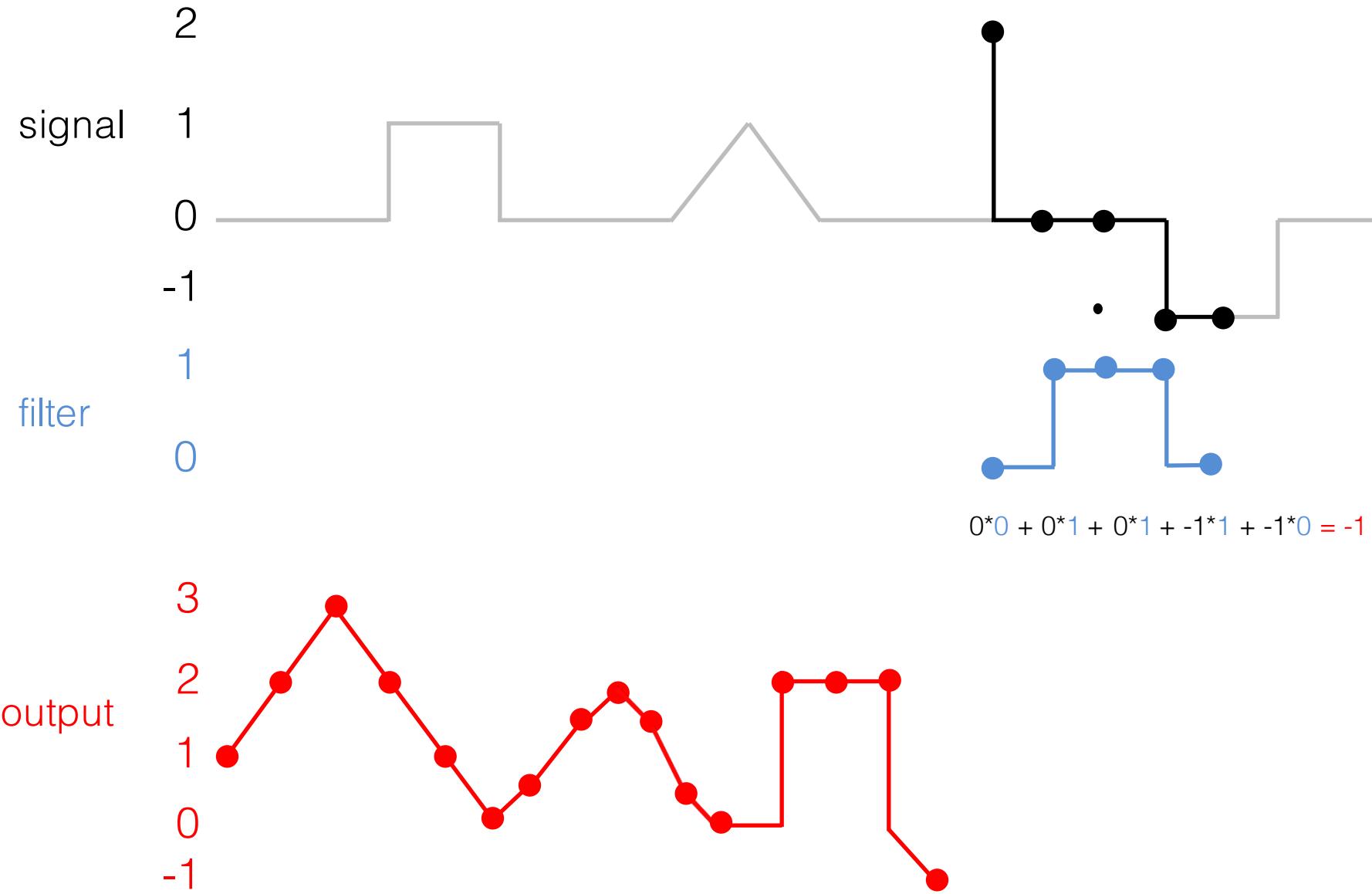
MATLAB: `conv(signal, filter, 'valid')`

Convolution

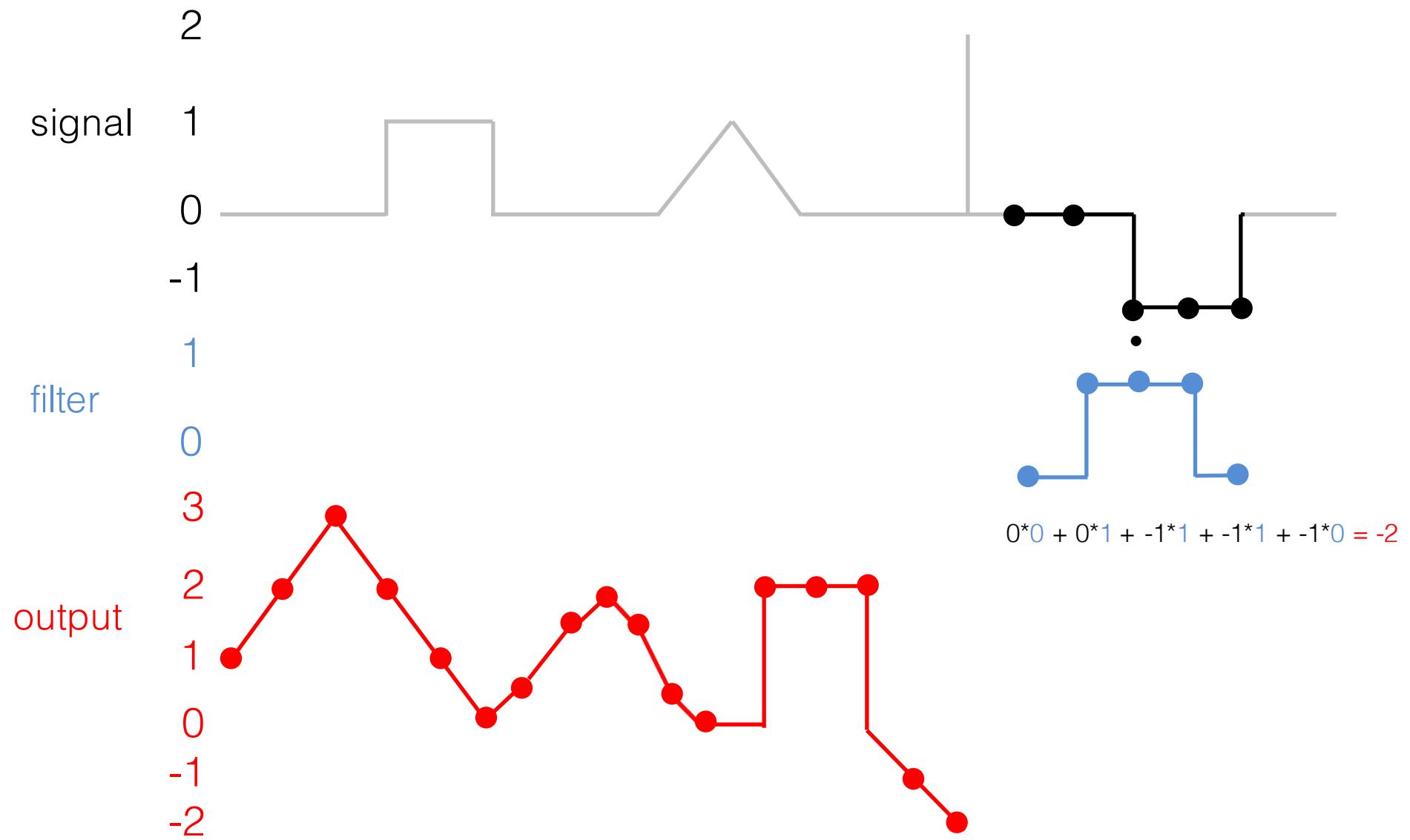


MATLAB: `conv(signal, filter, 'valid')`

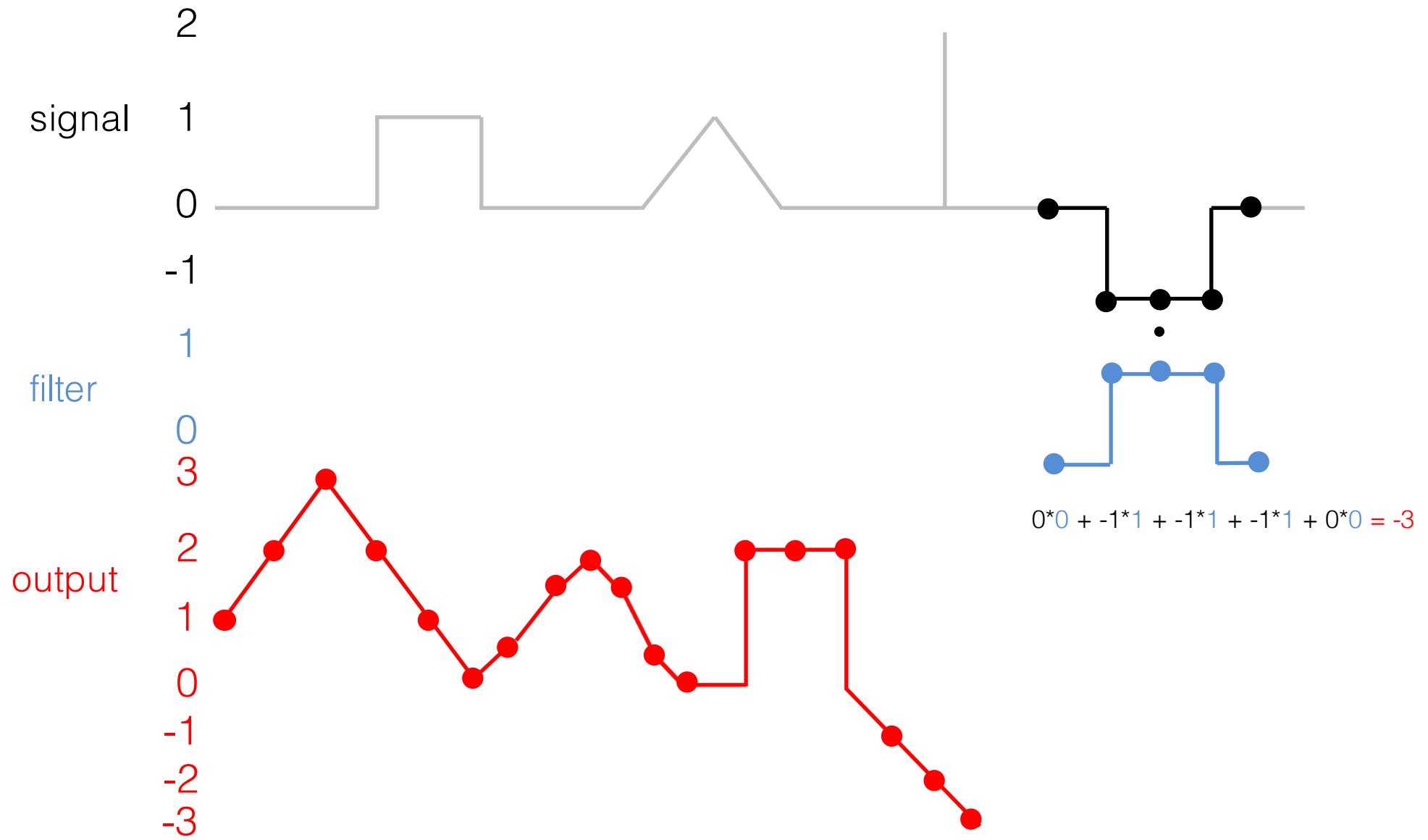
Convolution



Convolution

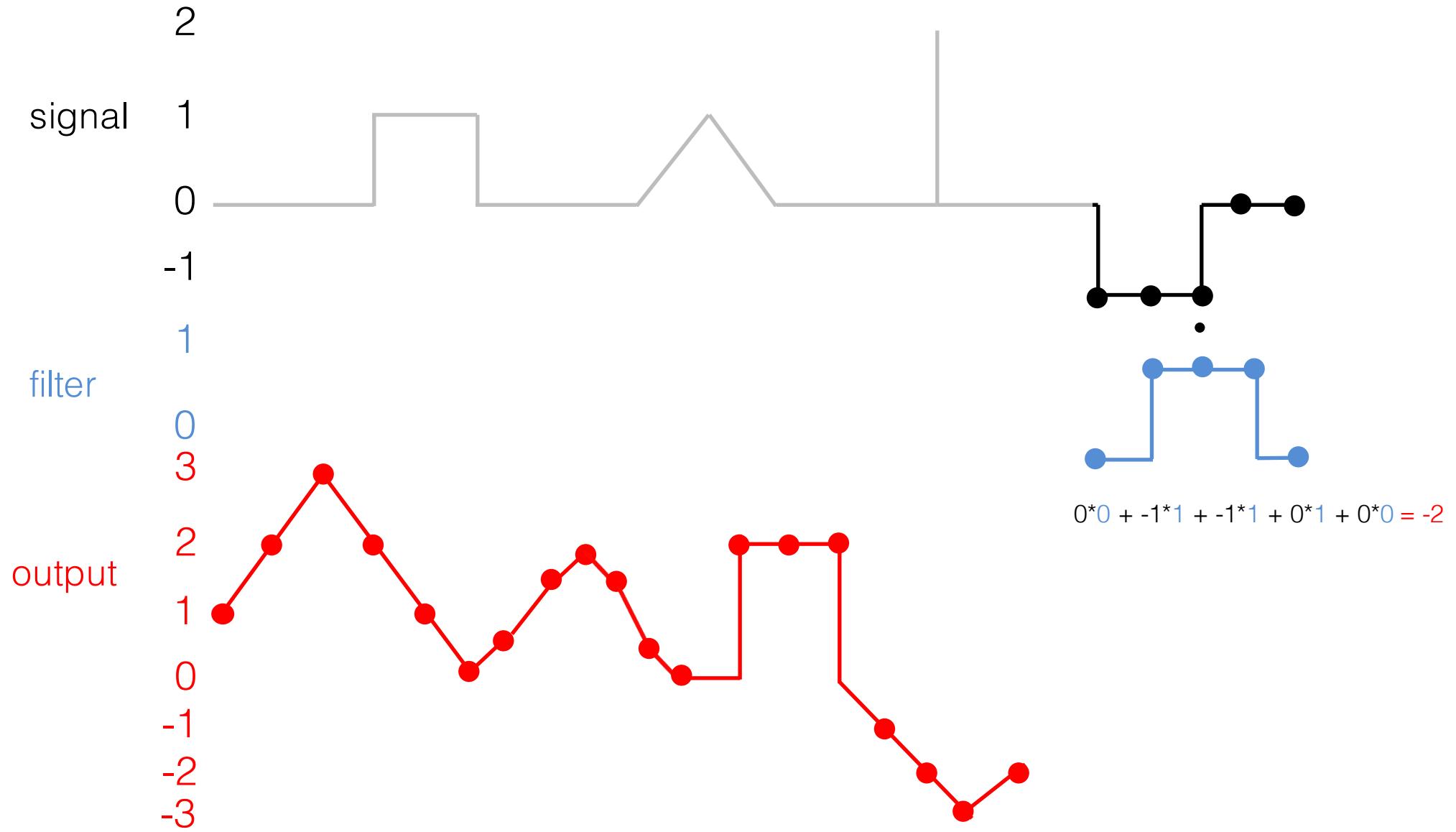


Convolution



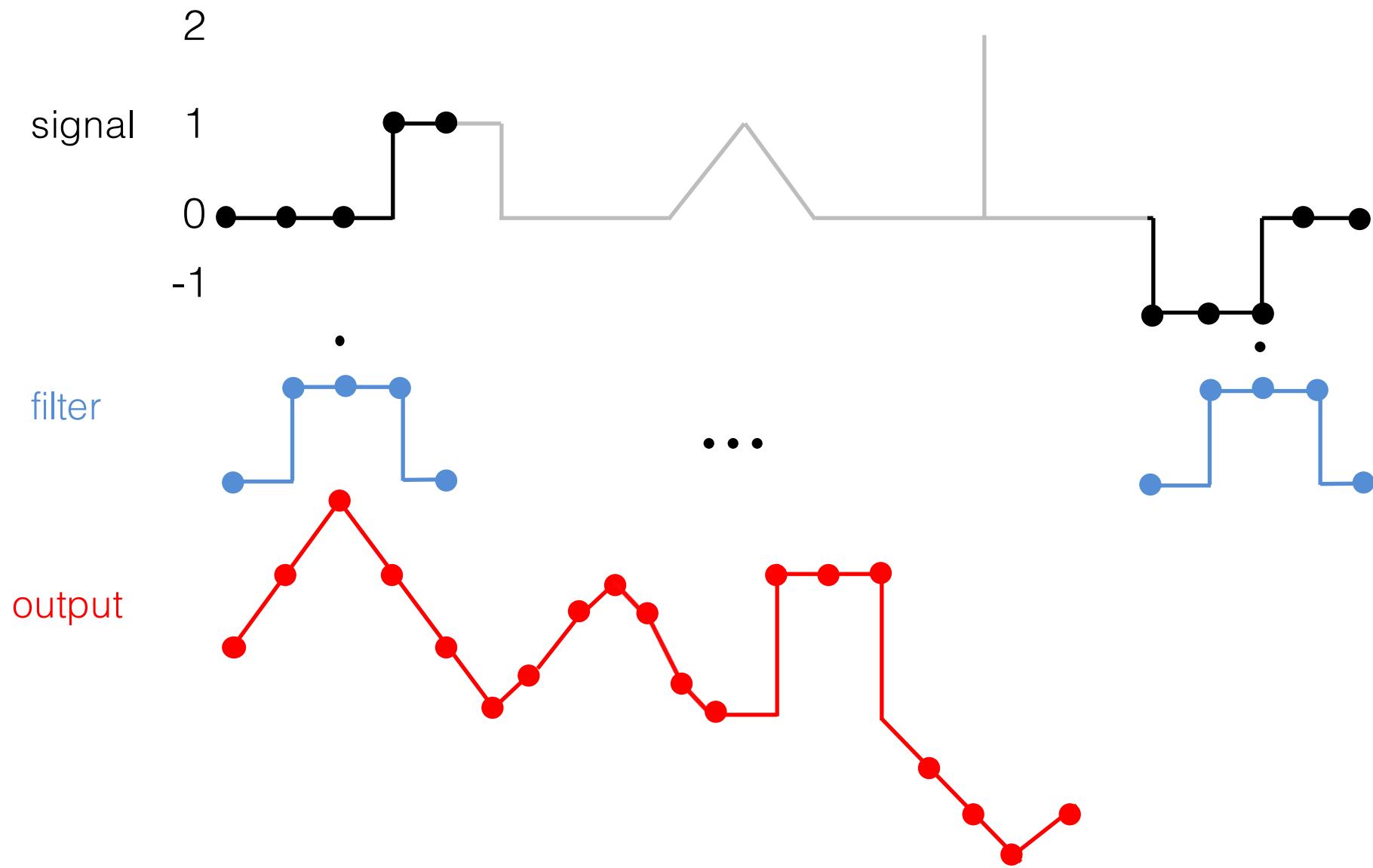
MATLAB: `conv(signal, filter, 'valid')`

Convolution



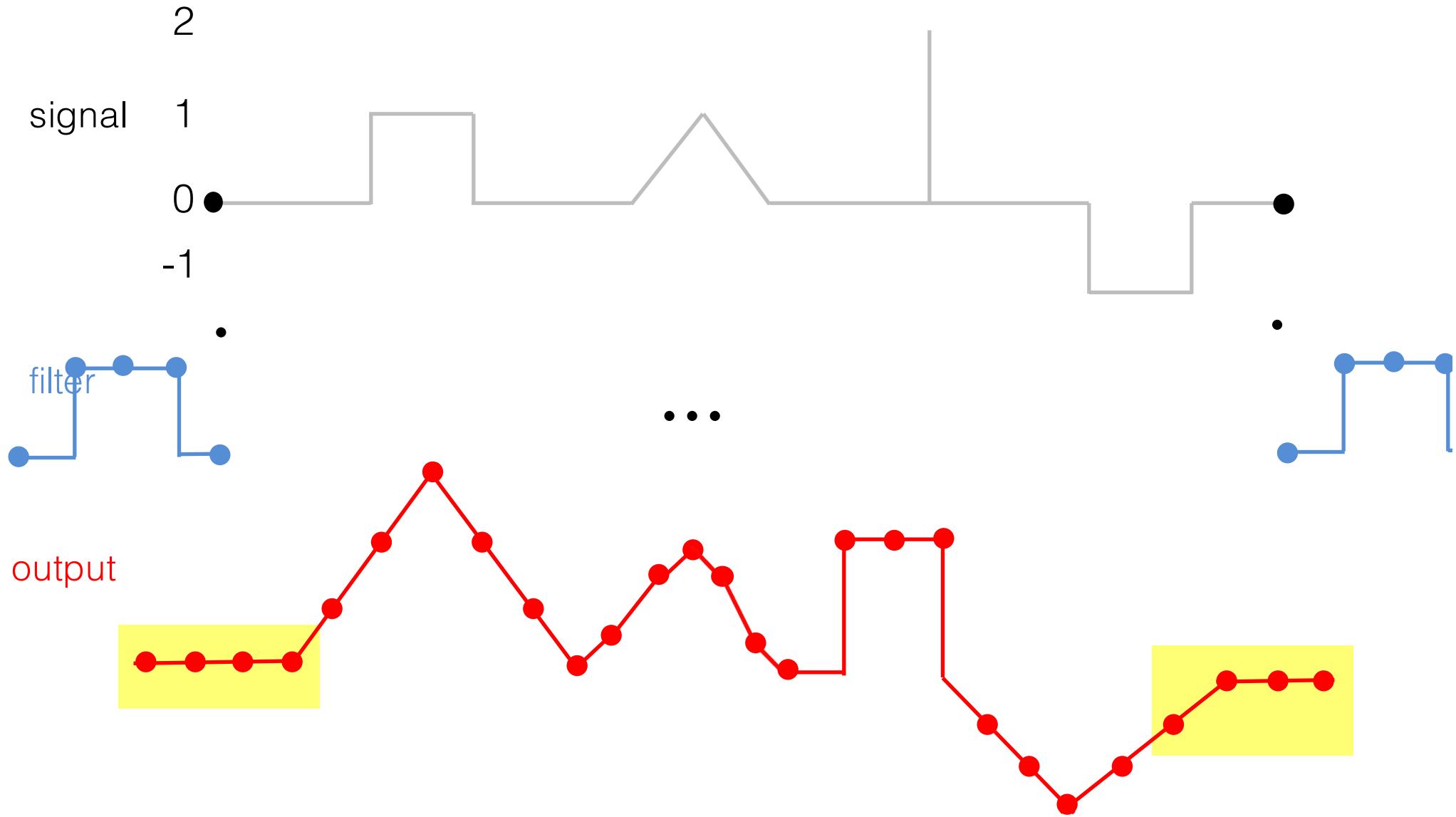
MATLAB: `conv(signal, filter, 'valid')`

Convolution



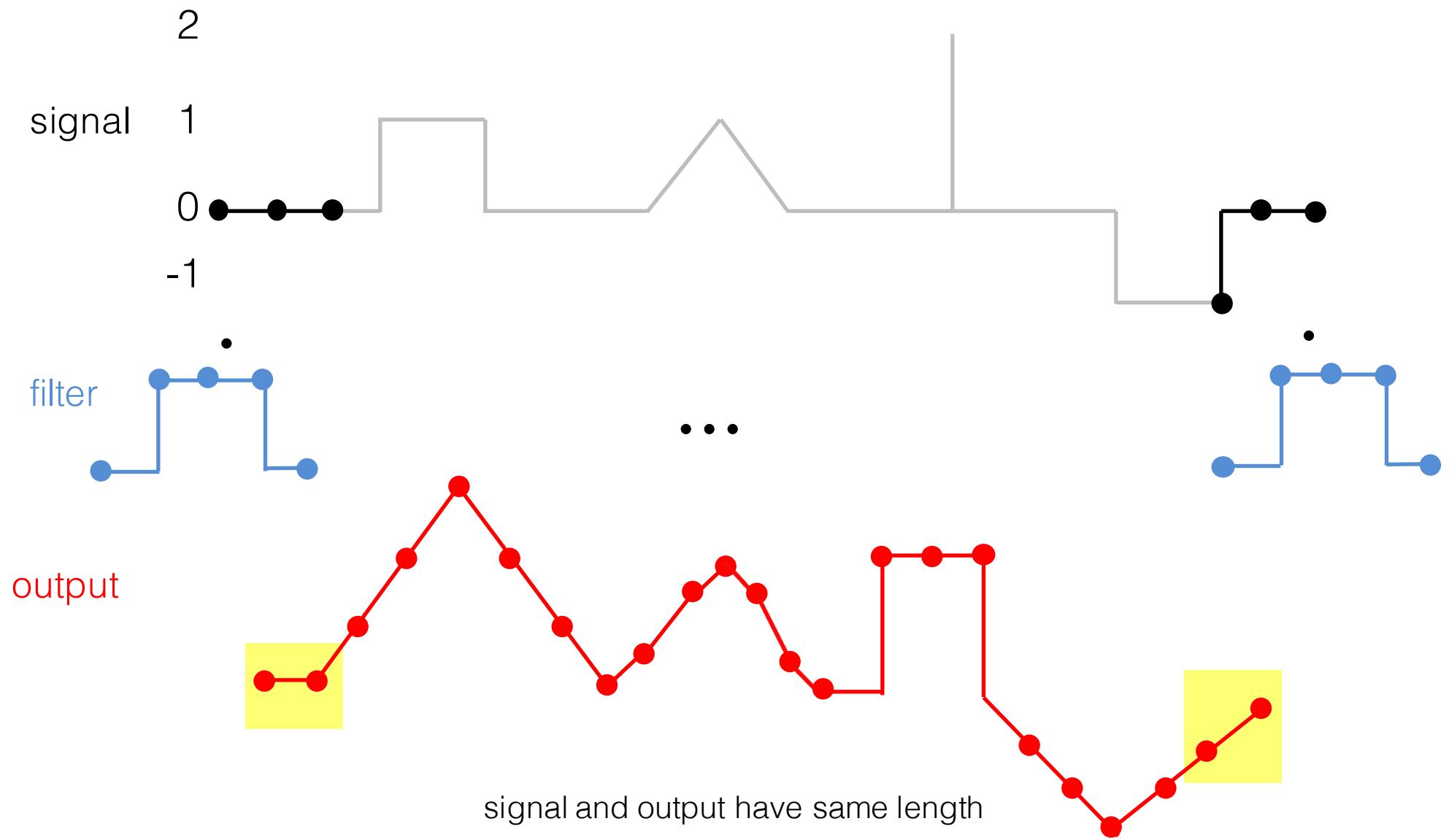
MATLAB: `conv(signal, filter, 'full')`

Convolution



MATLAB: `conv(signal, filter, 'same')`

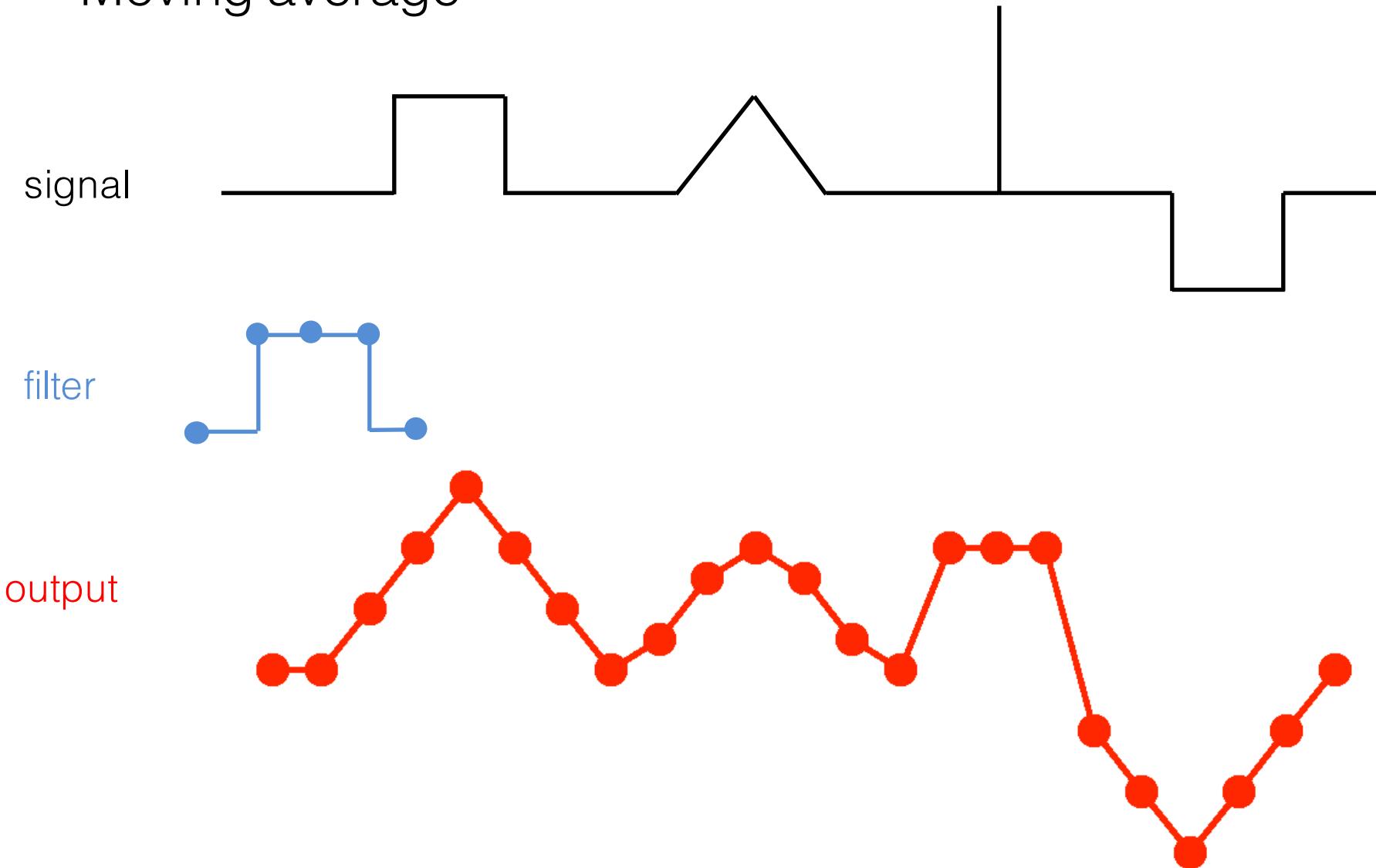
Convolution



MATLAB: `conv(signal, filter, 'same')`

Convolution examples

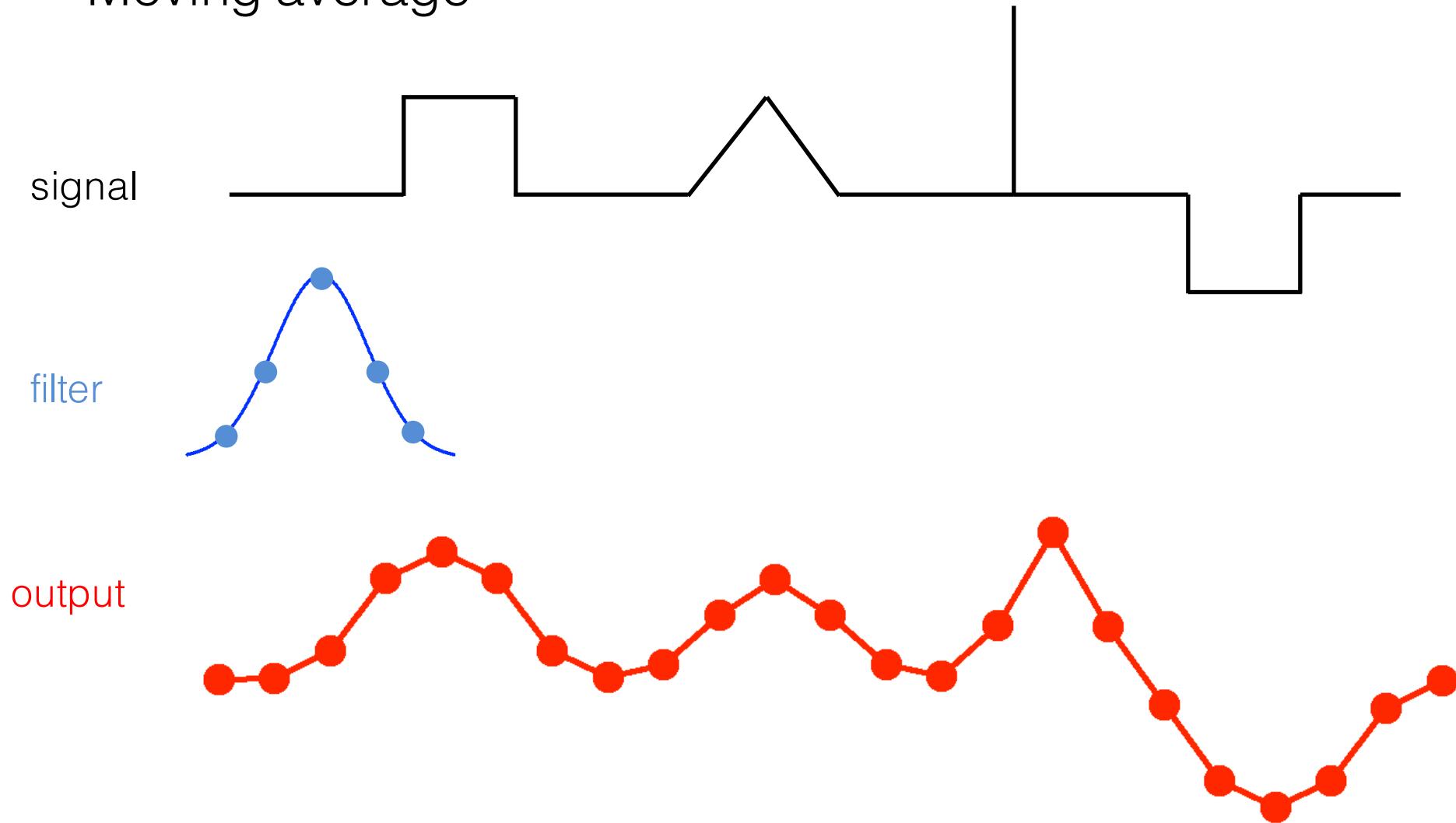
Moving average



MATLAB: `conv(signal, filter, 'same')`

Convolution examples

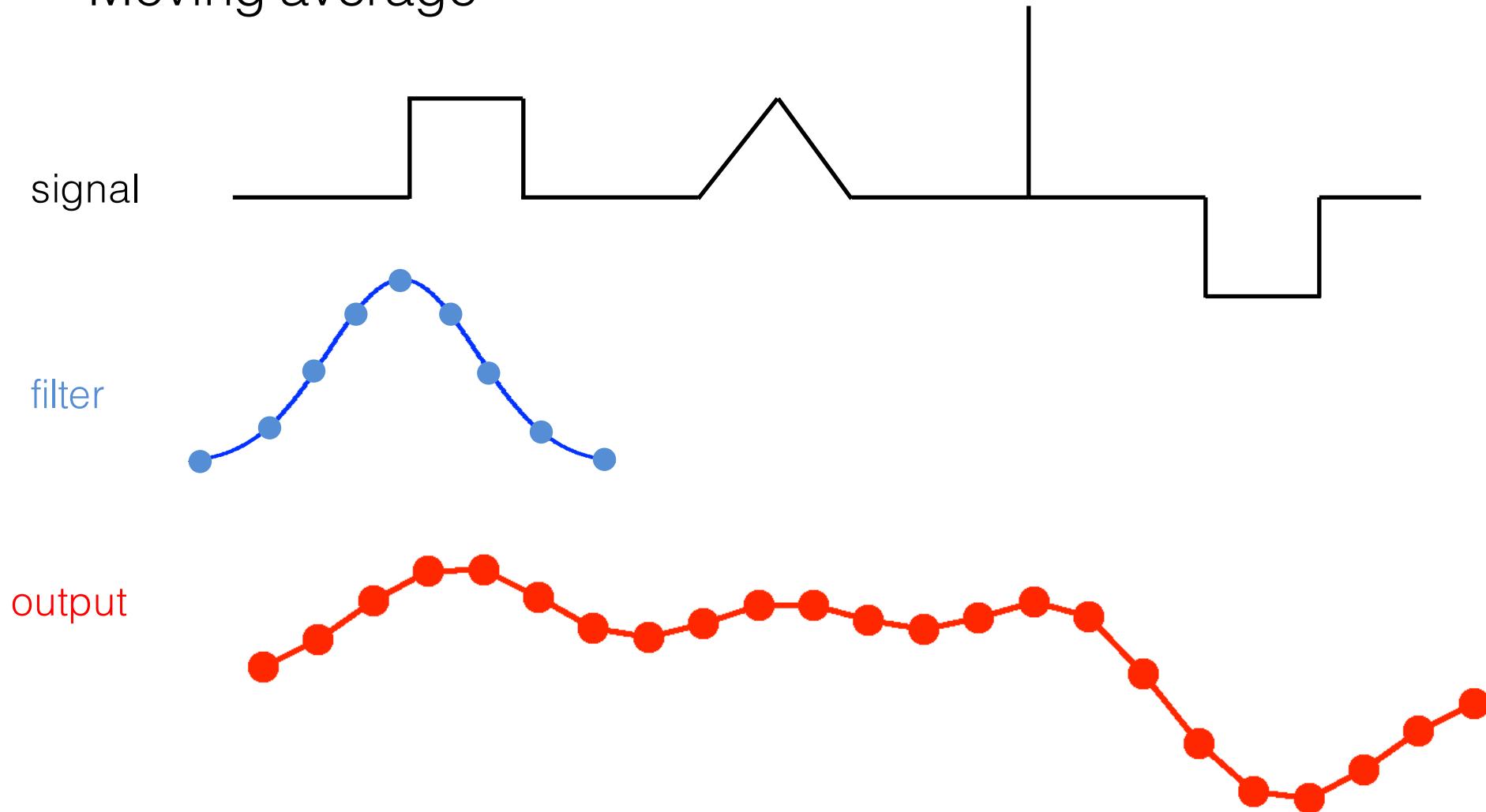
Moving average



MATLAB: `conv(signal, filter, 'same')`

Convolution examples

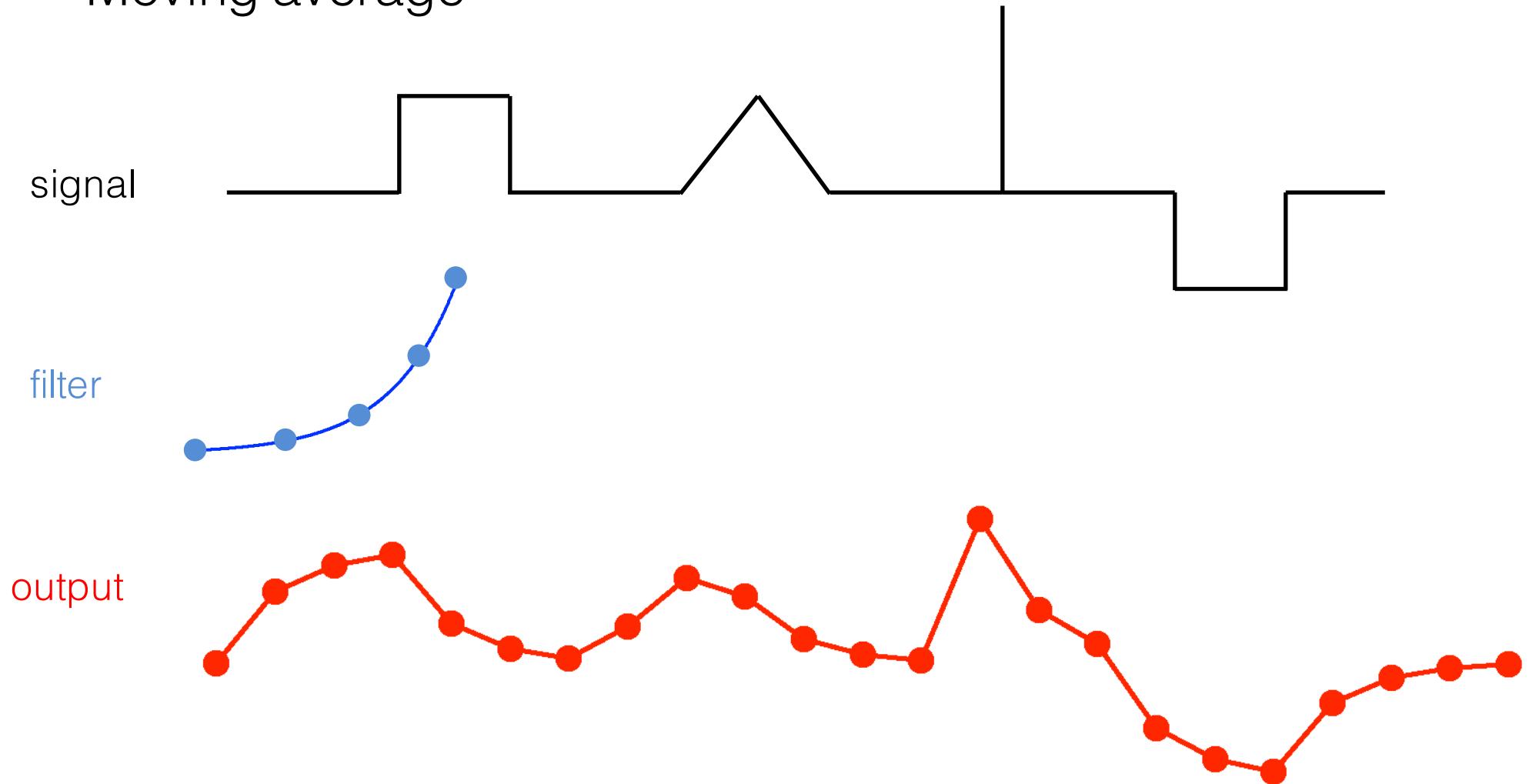
Moving average



MATLAB: `conv(signal, fliplr(filter), 'same')`

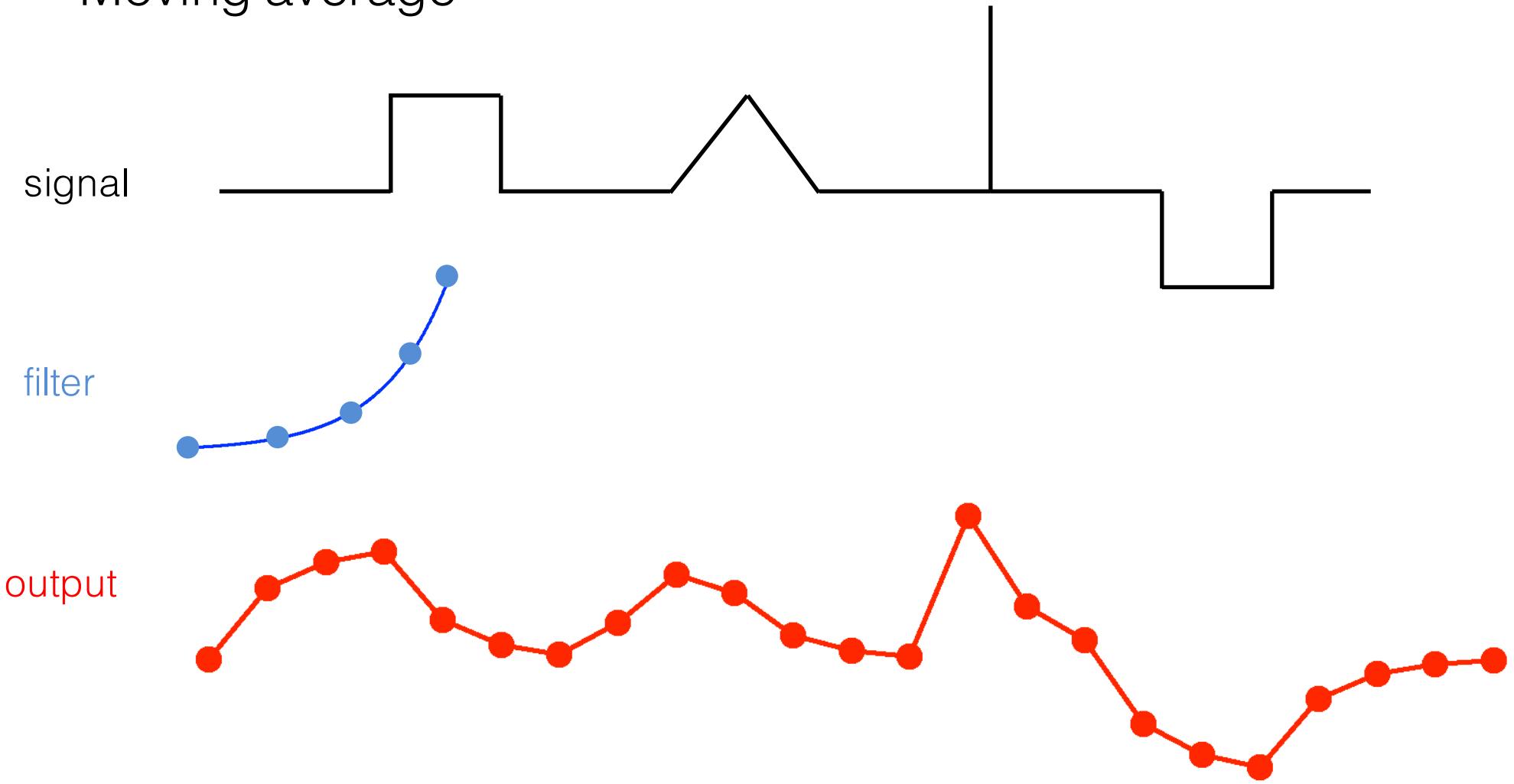
Convolution examples

Moving average



Convolution examples

Moving average

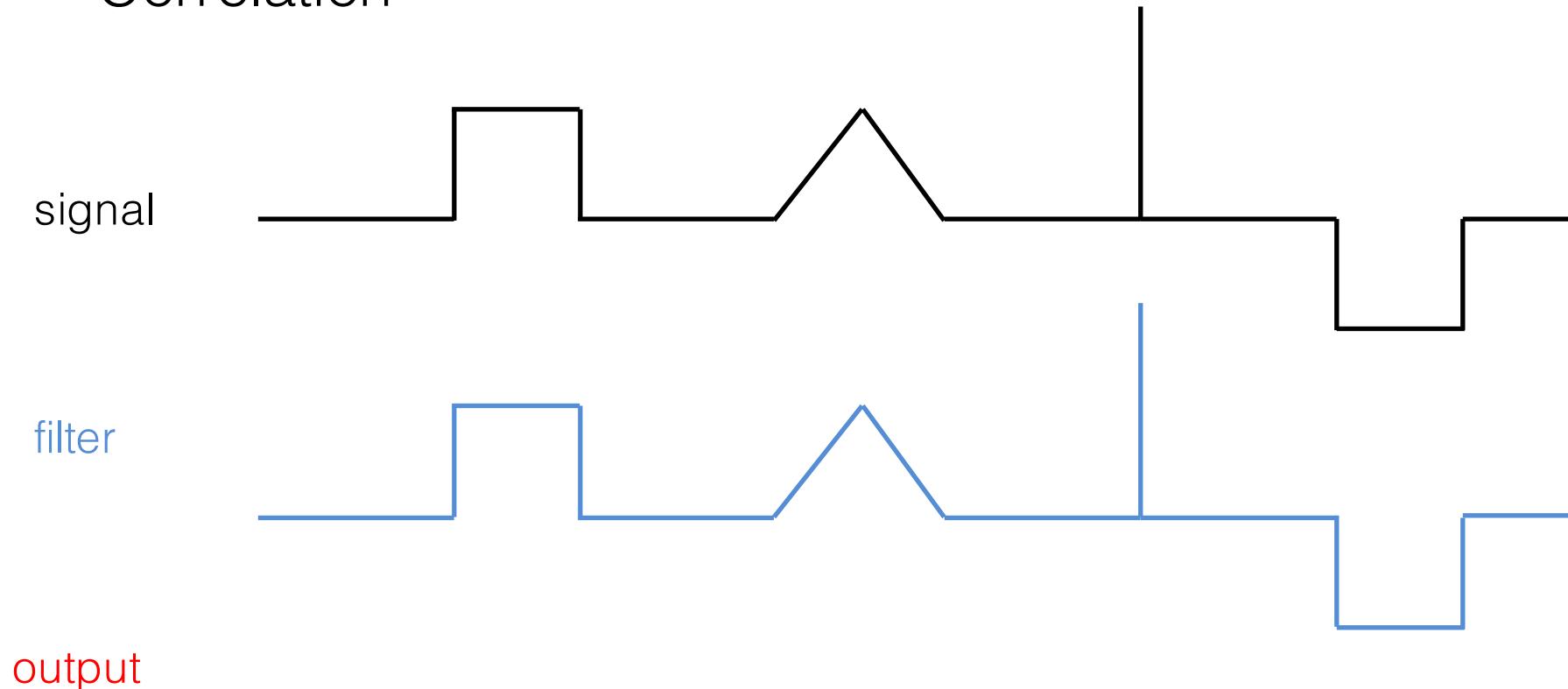


a note about using asymmetric filters in MATLAB

MATLAB: `conv(signal, fliplr(filter), 'same')`

Convolution examples

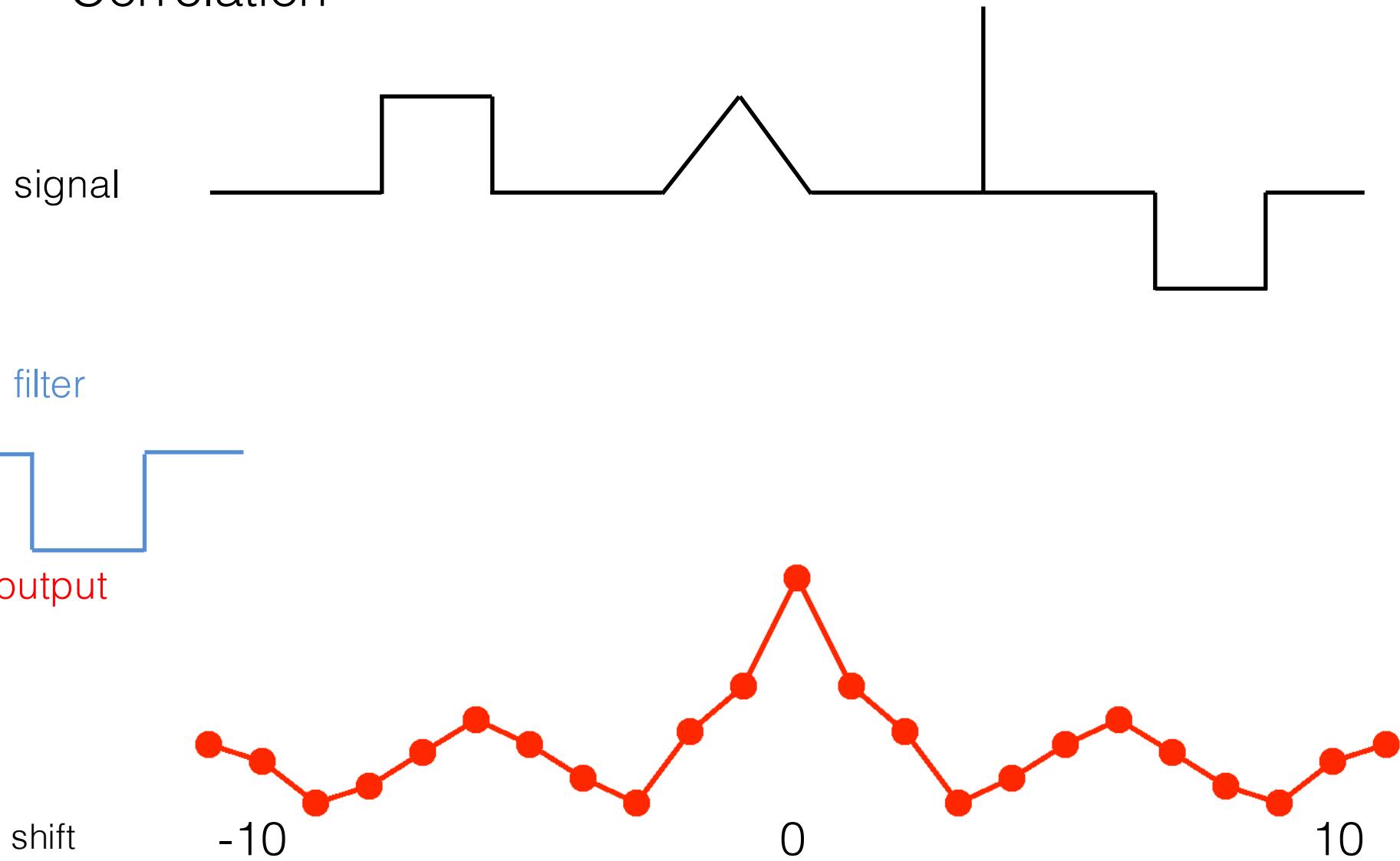
Correlation



MATLAB: `conv(signal, fliplr(filter), 'same')`

Convolution examples

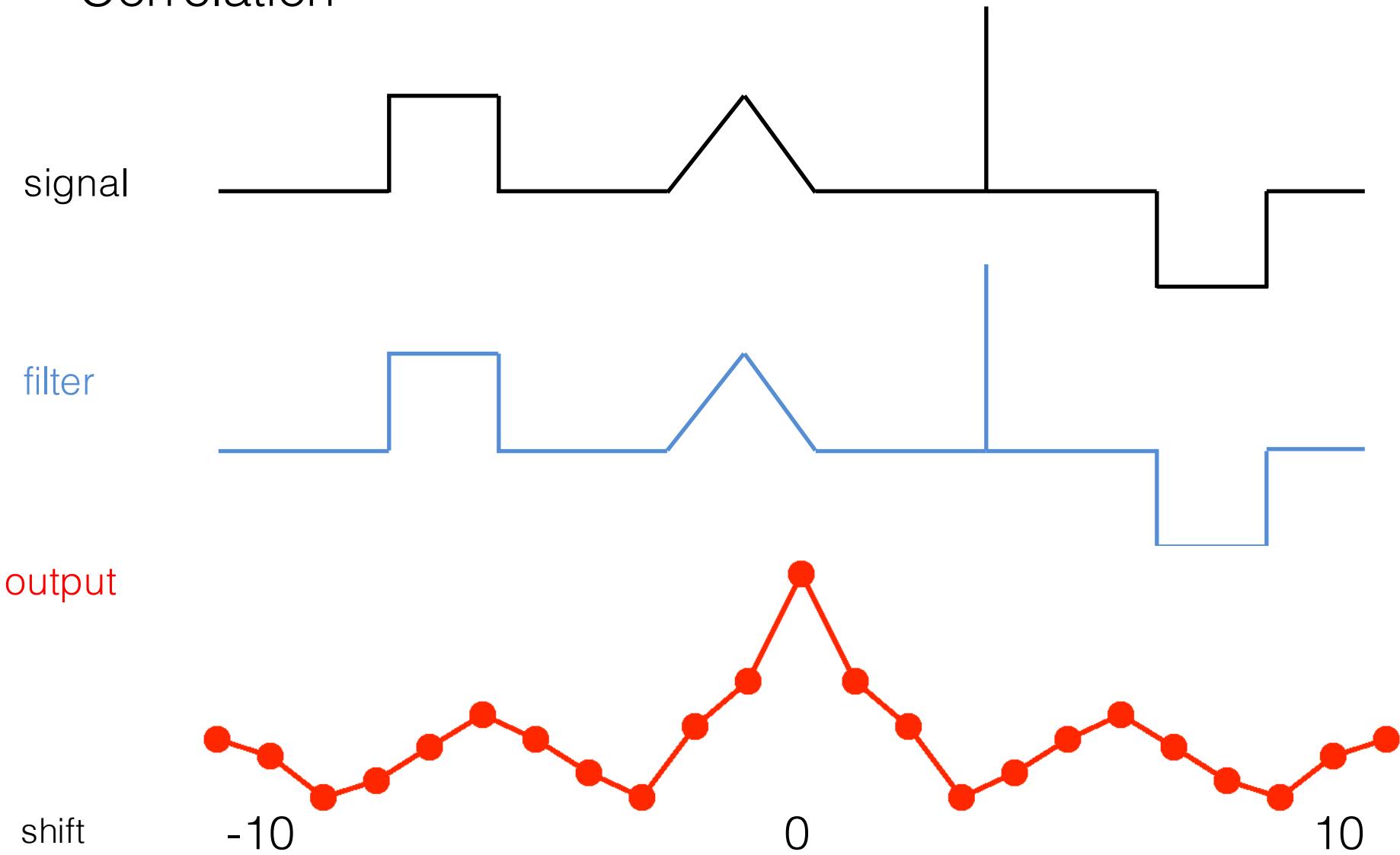
Correlation



MATLAB: `conv(signal, fliplr(filter), 'same')`

Convolution examples

Correlation



MATLAB: `conv2(signal, filter, 'same')`

2d Convolution examples

Image processing

signal



MATLAB: `conv2(signal, filter, 'same')`

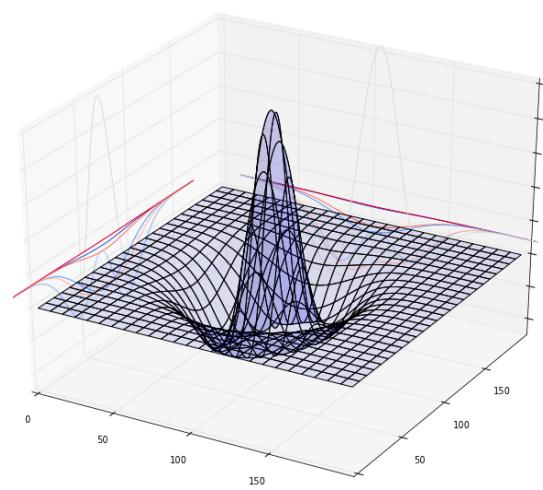
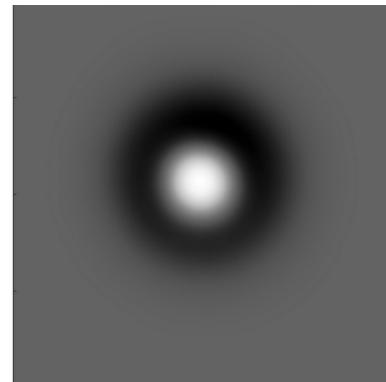
2d Convolution examples

Image processing

signal



filter



MATLAB: `conv2(signal, filter, 'same')`

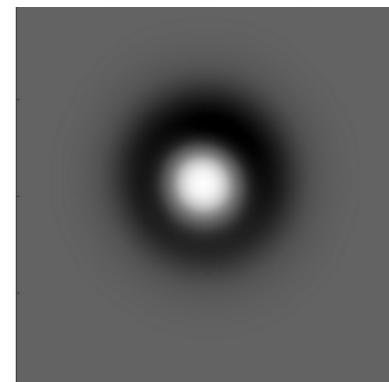
2d Convolution examples

Image processing

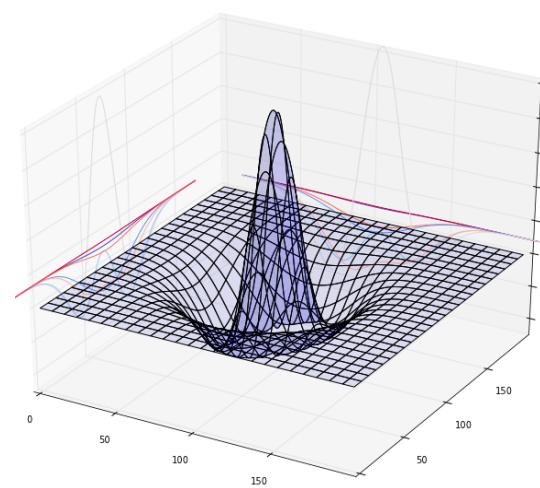
signal



filter



output



MATLAB: `conv2(signal, filter, 'same')`

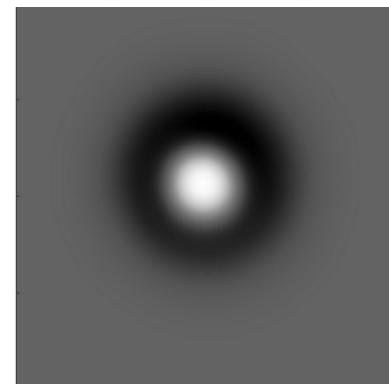
2d Convolution examples

Image processing

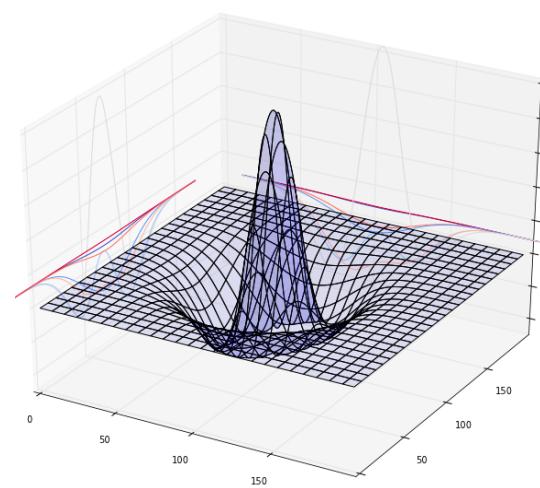
signal



filter



output



Question

If you try smoothing your dataset with a Gaussian of 10 ms but your output looks noisy, should you try convolving your stimulus with a

- a) wider Gaussian
- b) narrower Gaussian

Question

If you try smoothing your dataset with a Gaussian of 10 ms but your output looks noisy, should you try convolving your stimulus with a

- a) wider Gaussian

Question

To get the autocorrelation of a signal, you should do what first before convolving the signal with itself?

- a) flip the copied signal
- b) flip the copied signal and subtract the signal mean
- c) flip the copied signal and divide by the signal std
- d) find someone to help

Question

To get the autocorrelation of a signal, you should do what first before convolving the signal with itself?

- b) flip the copied signal and subtract the signal mean

Today's lecture

When will this be useful?

Filtering in time or space

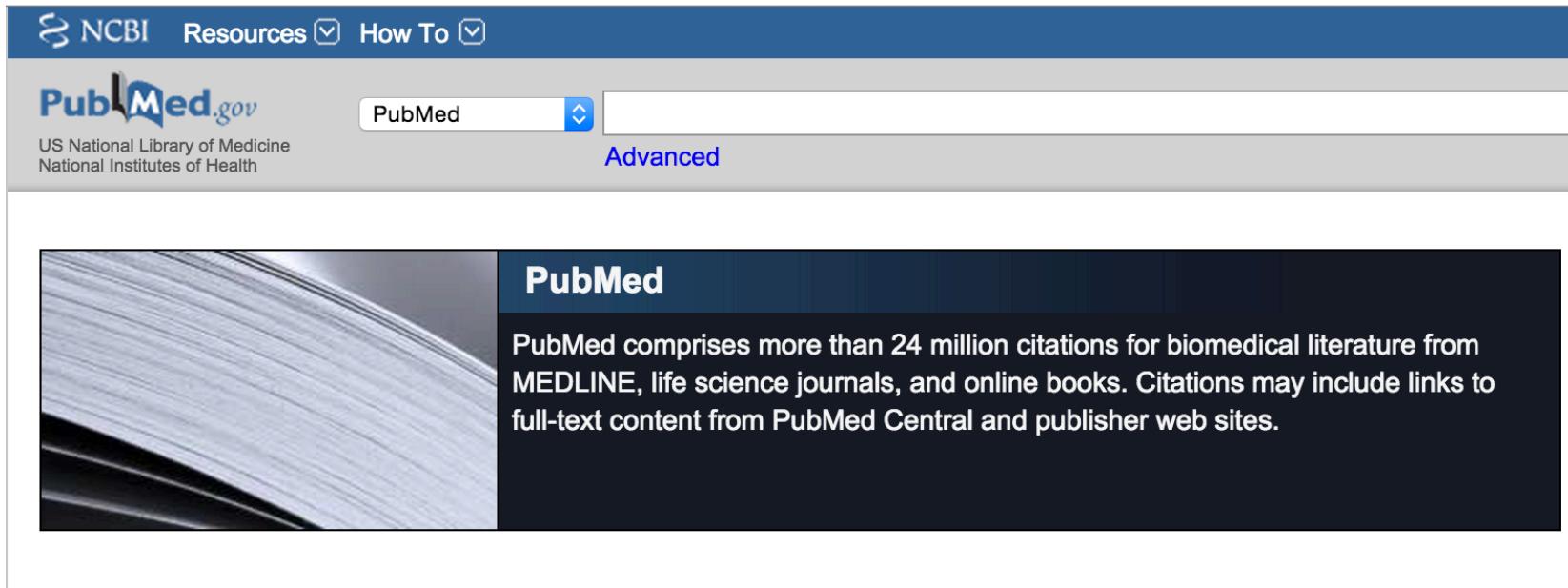
- What is convolution?
- How much should I filter?

Filtering in frequency

- Fourier series
- What is a Fourier transform?
- Designing filters in frequency domain

Fourier transform

Fourier transform

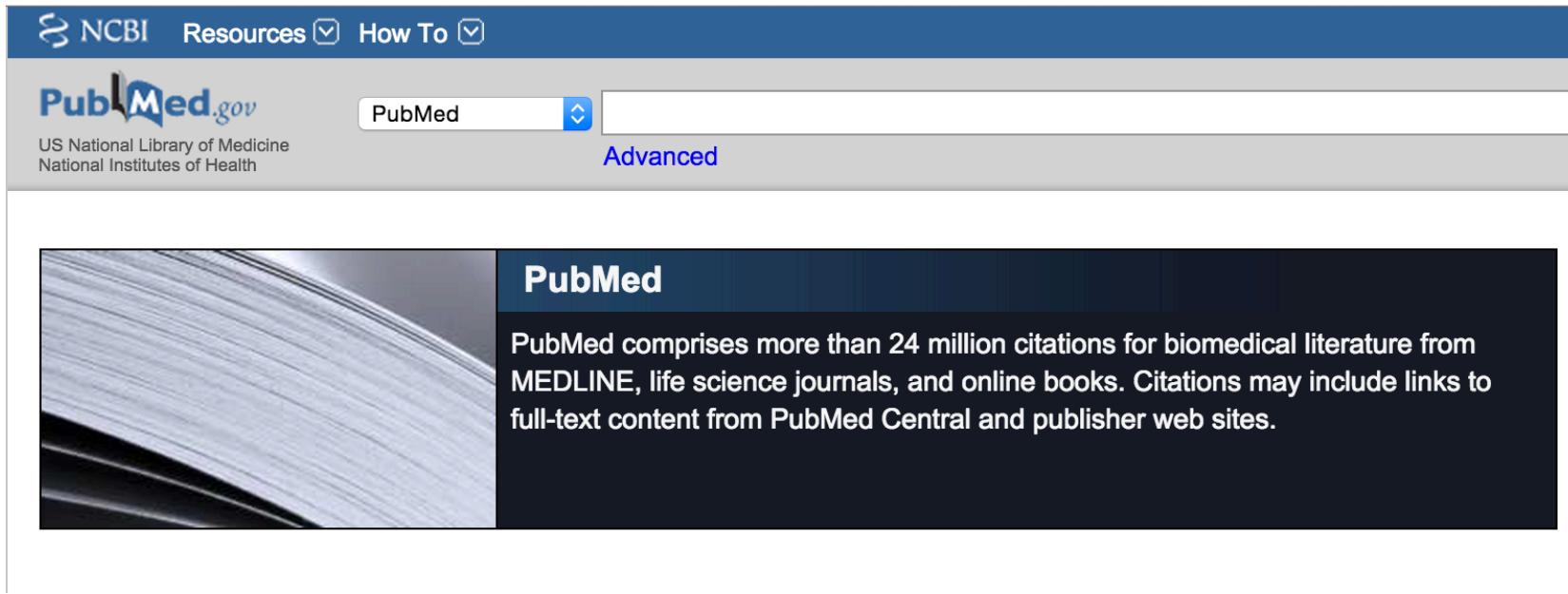


The screenshot shows the PubMed homepage. At the top, there is a blue header bar with the NCBI logo, a "Resources" dropdown menu, and a "How To" dropdown menu. Below the header is a grey navigation bar containing the PubMed logo, a search input field with the placeholder "PubMed", a dropdown arrow, and a link to "Advanced" search options. The main content area has a dark blue sidebar on the left featuring a grayscale image of a stack of papers. The main content area has a dark background and contains the following text:

PubMed

PubMed comprises more than 24 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher web sites.

Fourier transform



The screenshot shows the PubMed homepage. At the top, there's a blue header bar with the NCBI logo, a "Resources" dropdown, and a "How To" dropdown. Below the header is the PubMed logo and the text "US National Library of Medicine, National Institutes of Health". A search bar is centered, with "PubMed" selected in a dropdown menu and an "Advanced" link next to it. On the left, there's a large image of a stack of papers. To the right of the image, the word "PubMed" is written in white. Below that, a text box contains the following information: "PubMed comprises more than 24 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher web sites."

“Fourier” 79,577

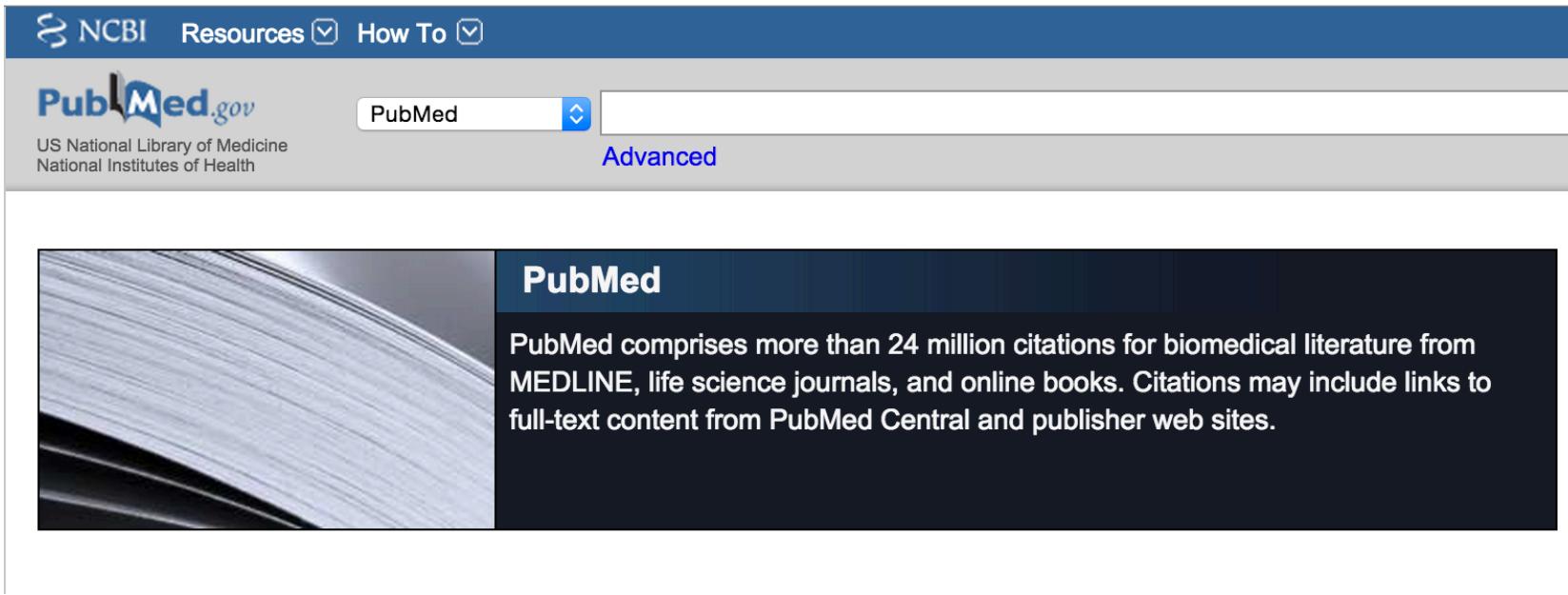
Fourier transform

The screenshot shows the PubMed homepage. At the top, there's a blue header bar with the NCBI logo, 'Resources' (with a dropdown arrow), and 'How To' (with a dropdown arrow). Below the header is the PubMed logo and the text 'US National Library of Medicine, National Institutes of Health'. A search bar is present with the word 'PubMed' and a dropdown arrow, and a link to 'Advanced' search. The main content area features a dark blue sidebar with a grayscale image of a stack of papers. The main text area is white with a dark blue header that reads 'PubMed'. It states: 'PubMed comprises more than 24 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher web sites.'

“Fourier” 79,577

“Filter” 58,593

Fourier transform



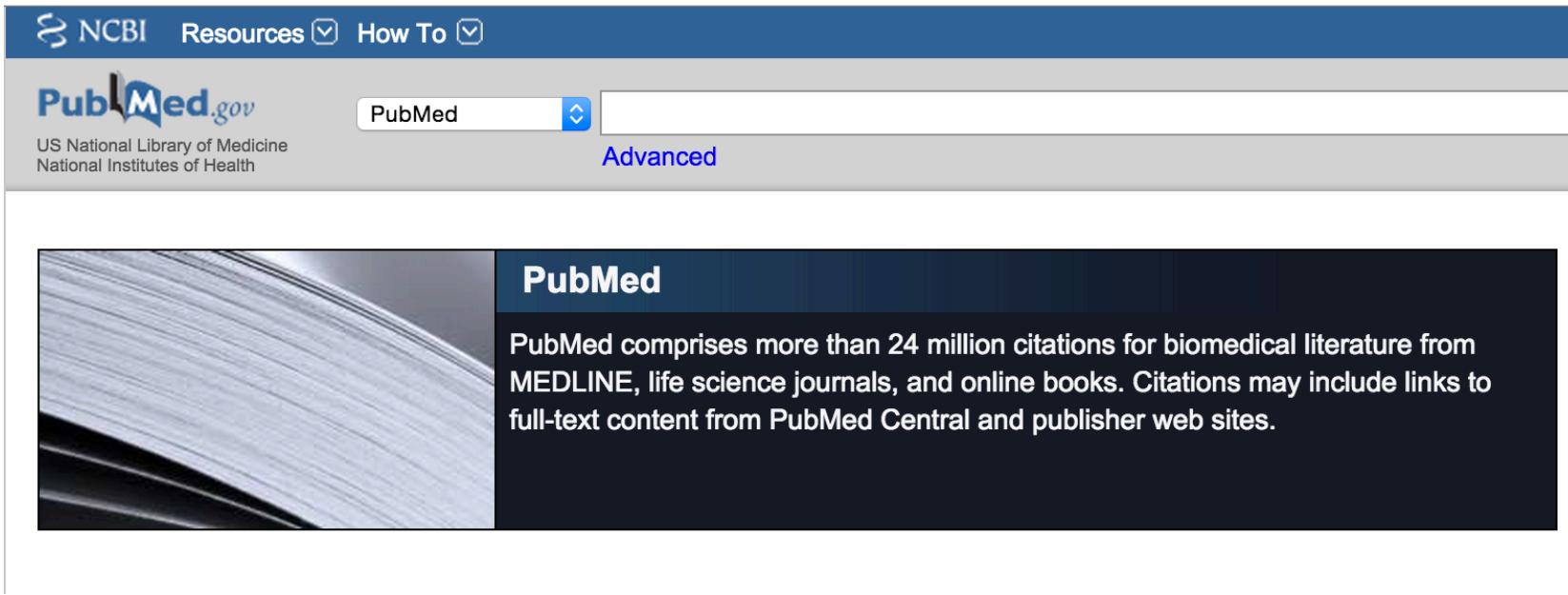
The screenshot shows the PubMed homepage. At the top, there's a blue header bar with the NCBI logo, a "Resources" dropdown, and a "How To" dropdown. Below the header is the PubMed logo and the text "US National Library of Medicine, National Institutes of Health". A search bar contains the word "PubMed" with a dropdown arrow, and a link to "Advanced" search. On the left, there's a large image of a stack of papers. On the right, a dark blue sidebar contains the text: "PubMed comprises more than 24 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher web sites."

“Fourier” 79,577

“Filter” 58,593

“Spectral analysis” 57,069

Fourier transform

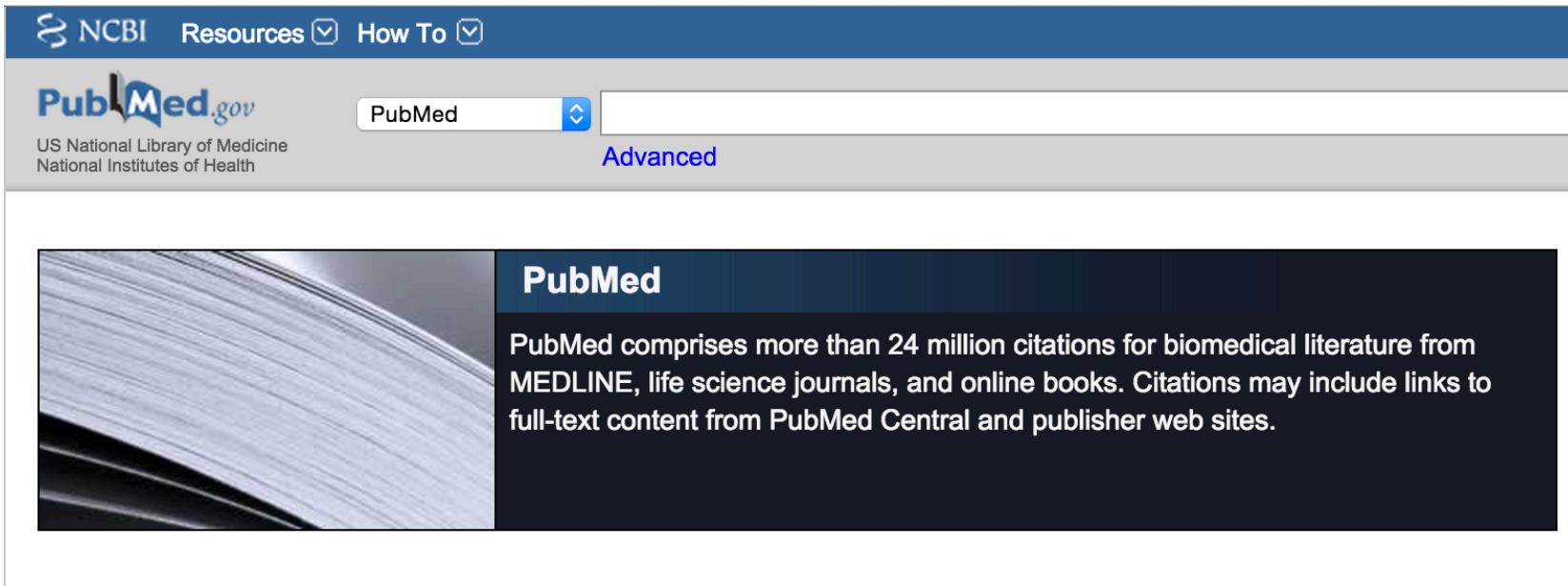


The screenshot shows the PubMed homepage. At the top, there's a blue header bar with the NCBI logo, a "Resources" dropdown, and a "How To" dropdown. Below the header is the PubMed logo and the text "US National Library of Medicine, National Institutes of Health". A search bar is centered, with "PubMed" selected in a dropdown menu and an "Advanced" link next to it. On the left side of the main content area is a large image of a stack of papers. To the right of the image, the word "PubMed" is written in white. Below this, a dark text box contains the following text: "PubMed comprises more than 24 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher web sites."

“Fourier”	79,577	“Drosophila”	92,215
“Filter”	58,593		
“Spectral analysis”	57,069		

as of 1/18/2016

Fourier transform



The screenshot shows the PubMed homepage. At the top, there's a blue header bar with the NCBI logo, a "Resources" dropdown, and a "How To" dropdown. Below the header is the PubMed logo and the text "US National Library of Medicine, National Institutes of Health". A search bar is centered, with "PubMed" selected in a dropdown menu and an "Advanced" link next to it. On the left, there's a large image of a stack of papers. To the right of the image, the word "PubMed" is written in white. Below this, a dark box contains the text: "PubMed comprises more than 24 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher web sites."

“Fourier”	79,577	“Drosophila”	92,215
“Filter”	58,593	“Elegans or <i>C. elegans</i> or <i>Caenorhabditis elegans</i> ”	27,396
“Spectral analysis”	57,069		

as of 1/18/2016

Other reasons Fourier transform is useful

Ease of computation

Filtering (convolution) is just element-wise multiplication in the frequency domain

Designing filters to remove noise is a lot easier in the frequency domain

Describing things

The cochlea transforms time domain signal into a frequency signal

Many brain regions have oscillations of a particular frequency

Fourier series

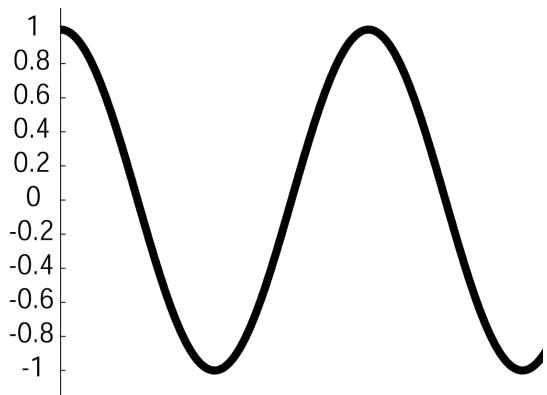
Every signal is equal to a sum of sines and cosines

Fourier series

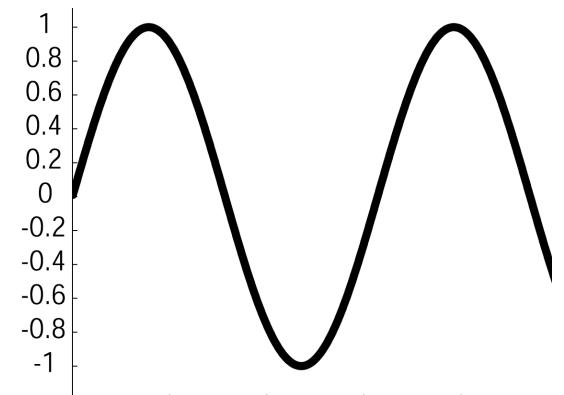
Every signal is equal to a sum of sines and cosines

$$a \cos(2\pi f x) + b \sin(2\pi f x)$$

cosine



sine

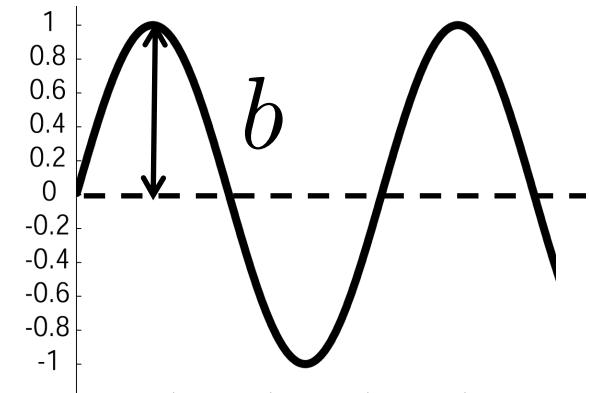
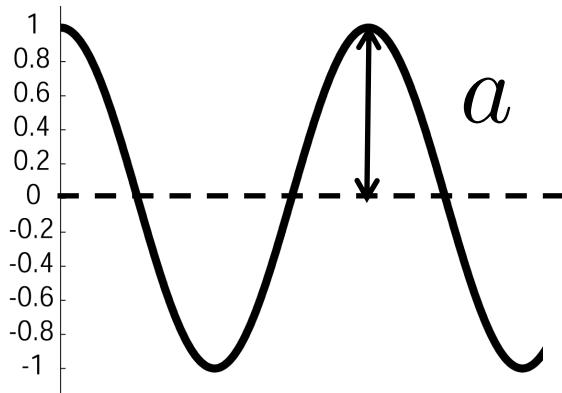


Fourier series

Every signal is equal to a sum of sines and cosines

$$a \cos(2\pi f x) + b \sin(2\pi f x)$$

coefficients (amplitudes)



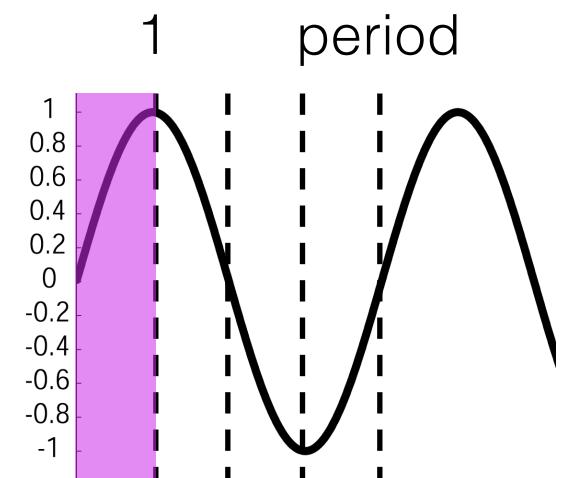
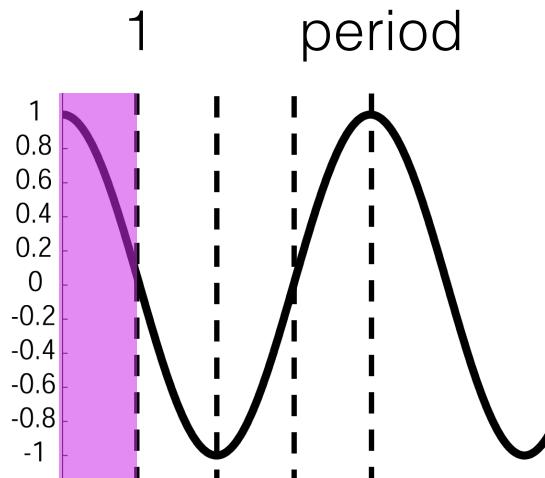
Fourier series

Every signal is equal to a sum of sines and cosines

$$a \cos(2\pi f x) + b \sin(2\pi f x)$$

coefficients (amplitudes)

frequencies



Fourier series

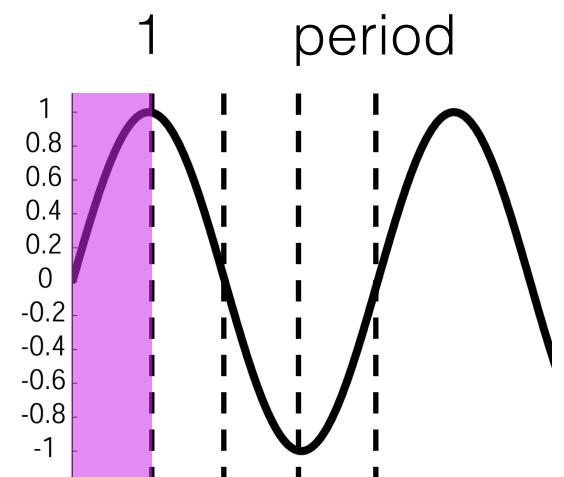
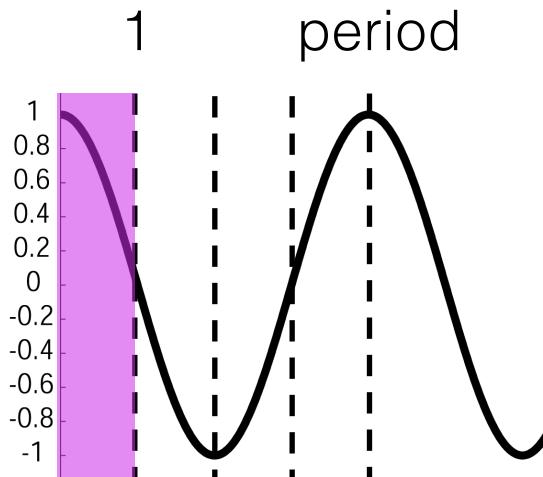
Every signal is equal to a sum of sines and cosines

$$a \cos(2\pi f x) + b \sin(2\pi f x)$$

$$f = 1/4$$

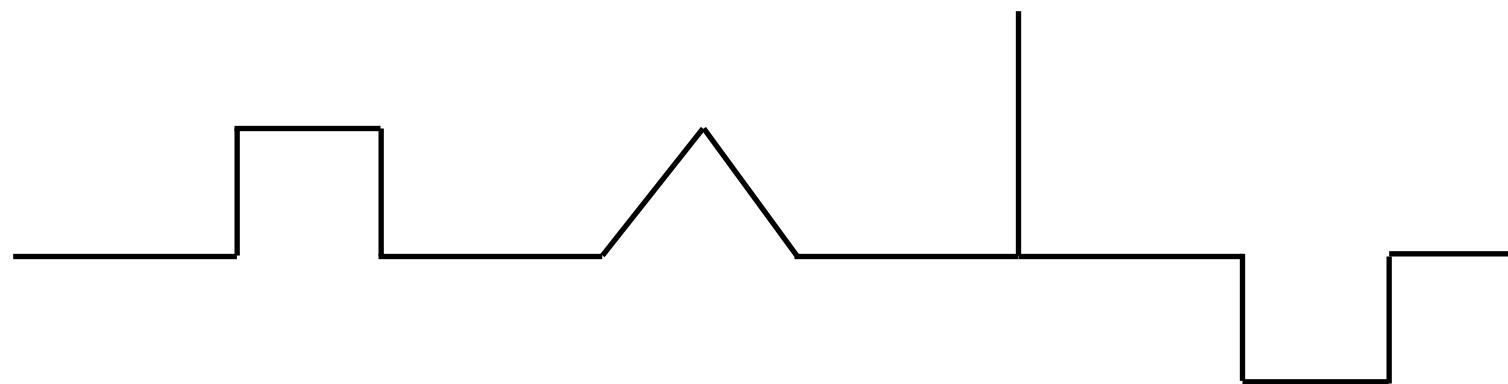
coefficients (amplitudes)

$$f = 1/4$$

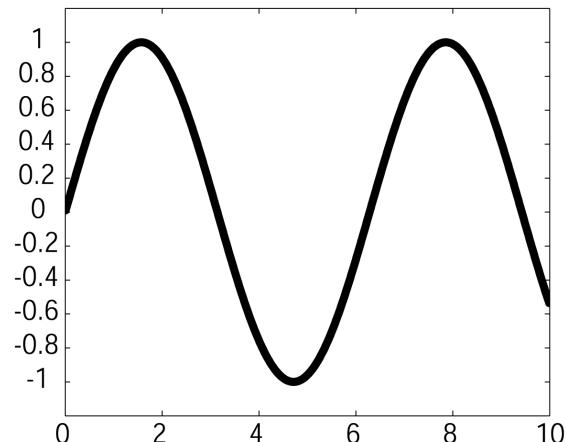


Fourier series

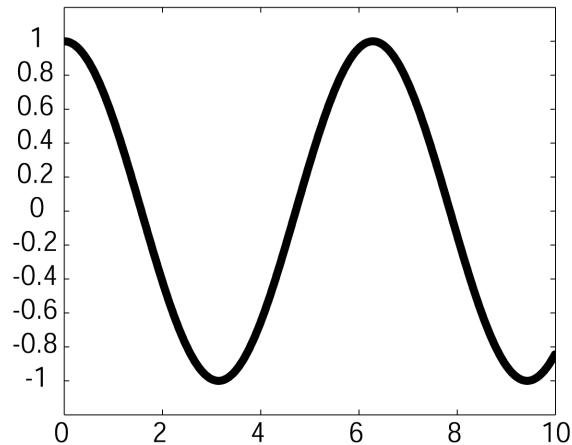
Every signal is equal to a sum of sines and cosines



sine (amplitude 1, period 2π)



cosine (amplitude 1, period 2π)

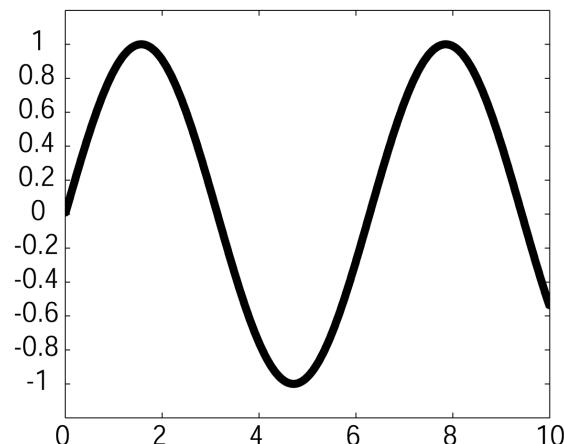


Fourier series

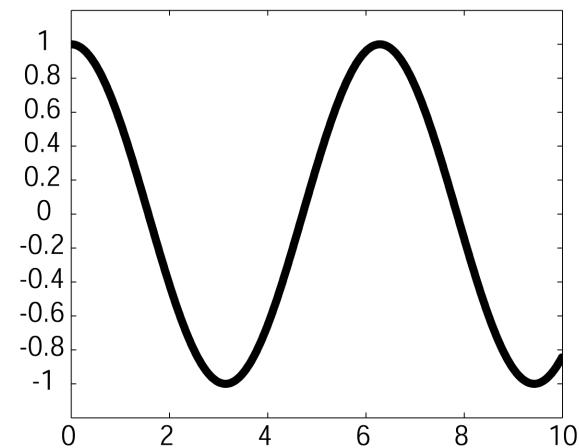
Every signal is equal to a sum of sines and cosines



sine (amplitude 1, period 2π)

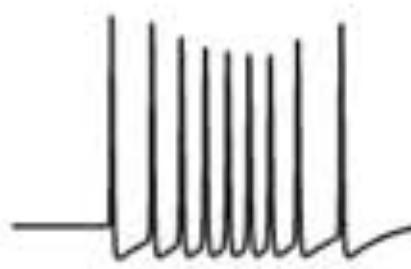


cosine (amplitude 1, period 2π)

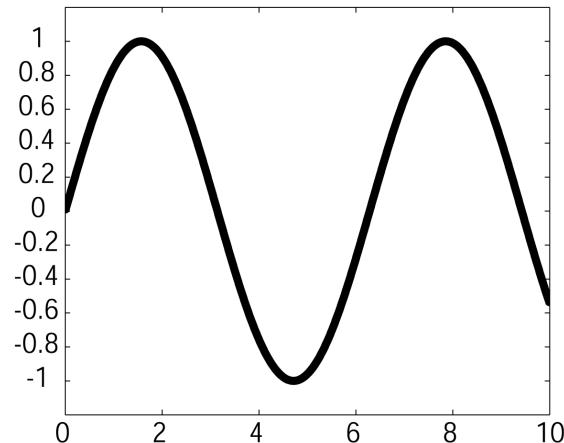


Fourier series

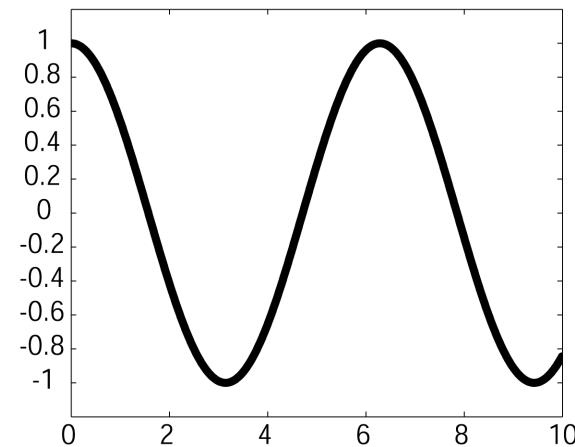
Every signal is equal to a sum of sines and cosines



sine (amplitude 1, period 2π)

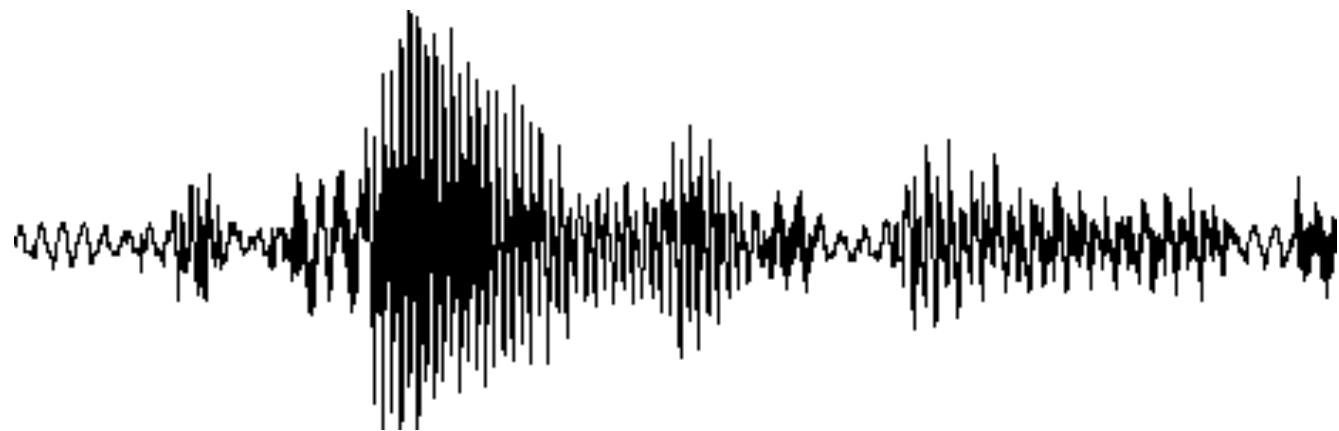


cosine (amplitude 1, period 2π)

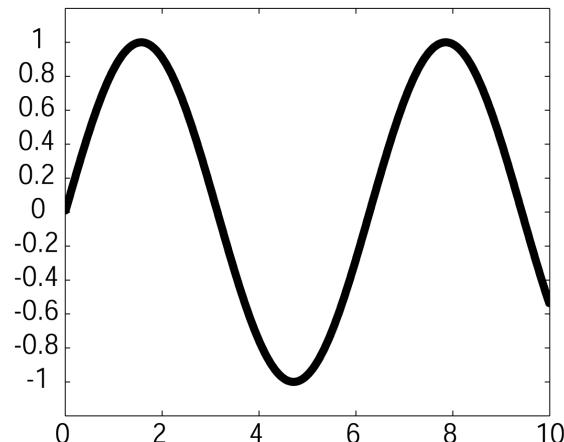


Fourier series

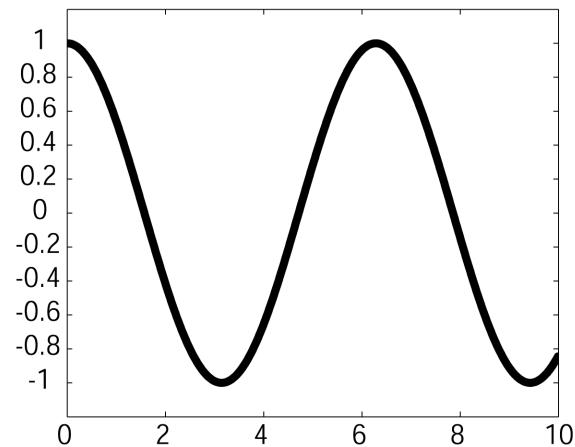
Every signal is equal to a sum of sines and cosines



sine (amplitude 1, period 2π)



cosine (amplitude 1, period 2π)



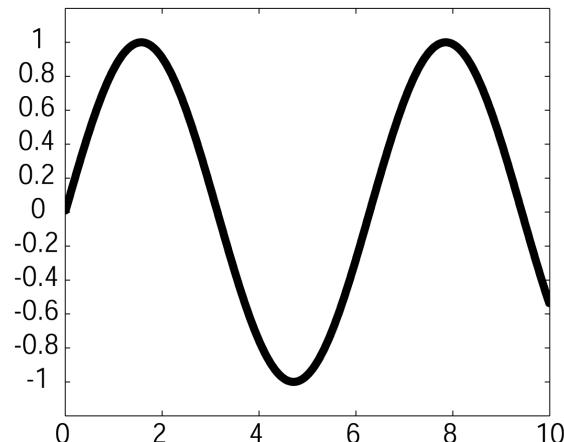
How is this possible?

Let's find the “Fourier series” representation of a simple signal

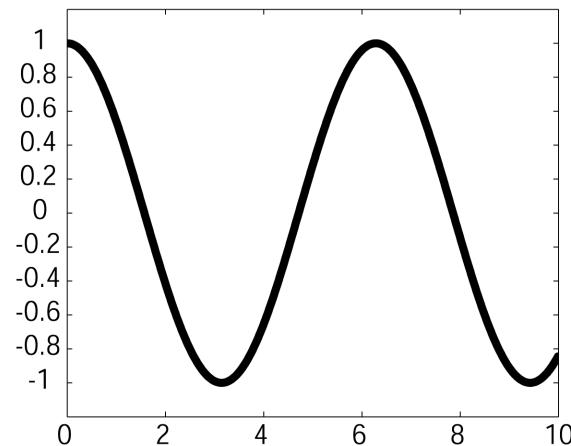
2

-1 ●

sine (amplitude 1, period 2π)

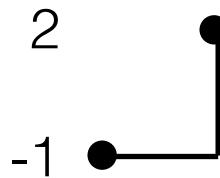


cosine (amplitude 1, period 2π)

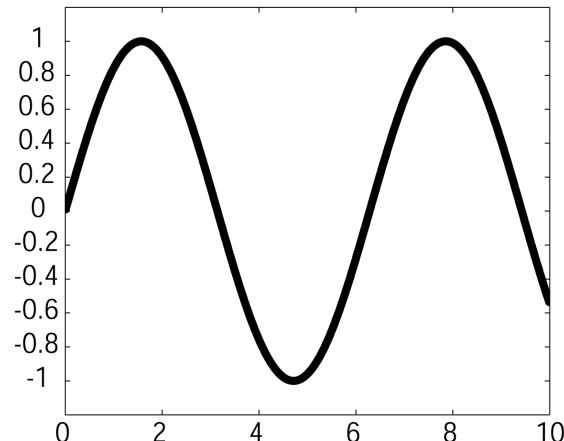


How is this possible?

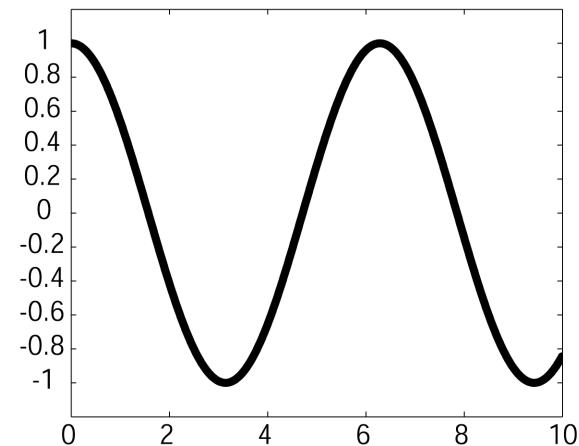
Let's find the “Fourier series” representation of a simple signal



sine (amplitude 1, period 2π)

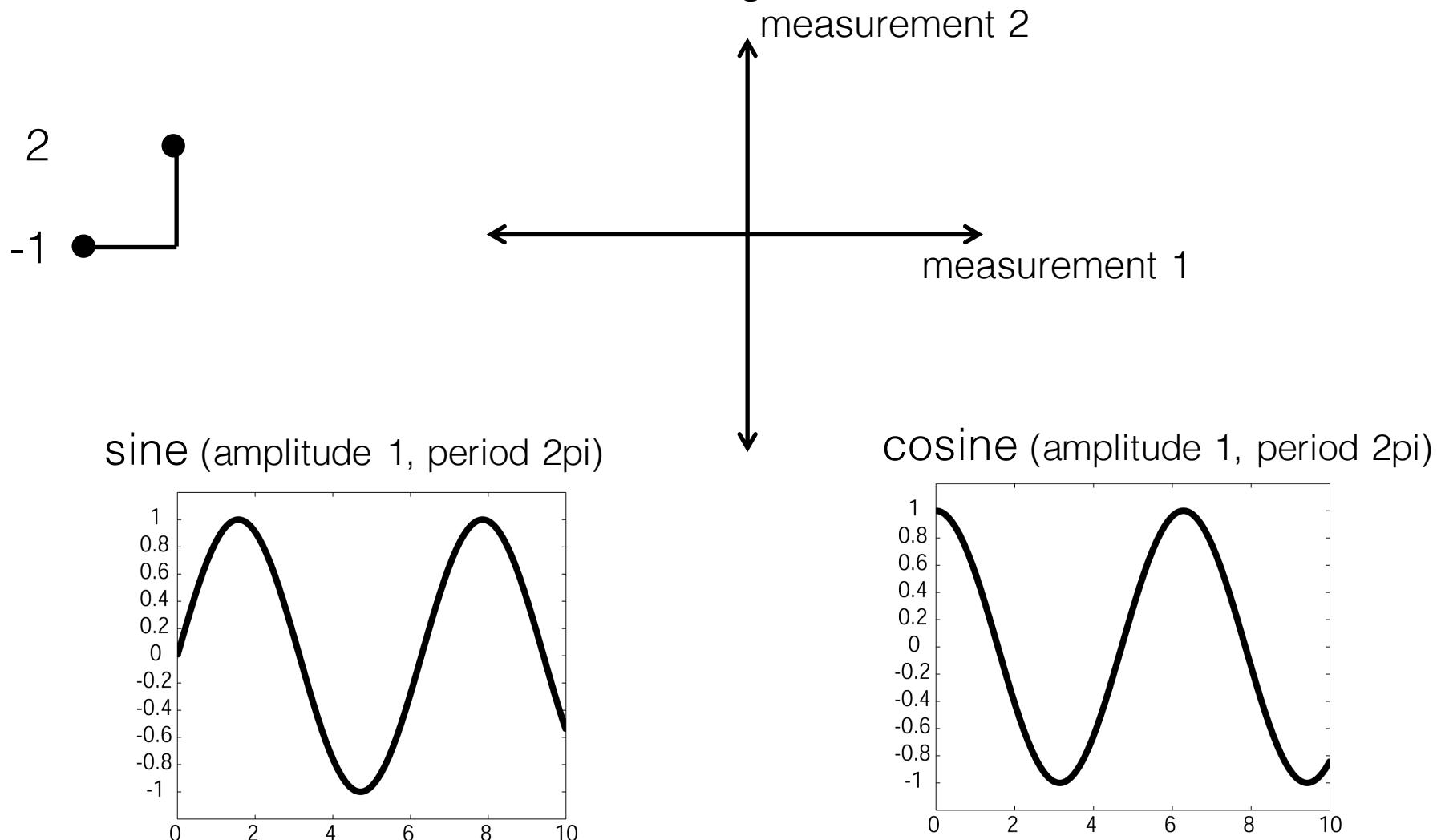


cosine (amplitude 1, period 2π)



How is this possible?

Let's find the “Fourier series” representation of a simple signal

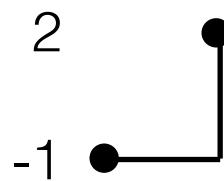


How is this possible?

Let's find the "Fourier series" representation of a simple

signal

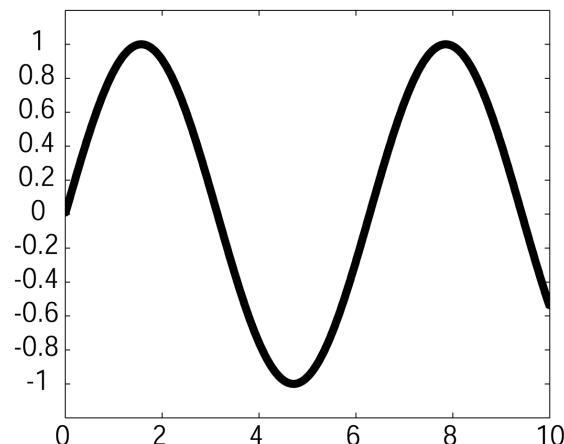
$$\begin{pmatrix} -1 \\ 2 \end{pmatrix}$$



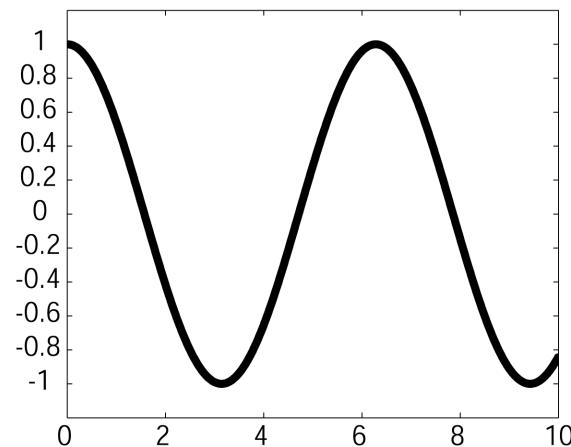
← measurement 1

measurement 2

sine (amplitude 1, period 2pi)



cosine (amplitude 1, period 2pi)

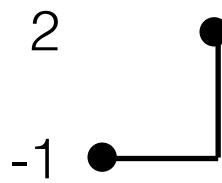


How is this possible?

Let's find the "Fourier series" representation of a simple

signal

$$\begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

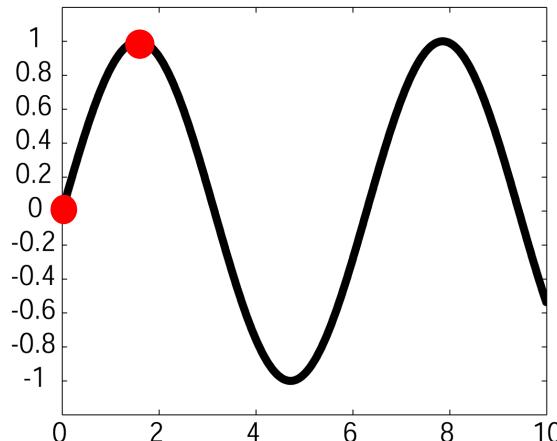


← measurement 1

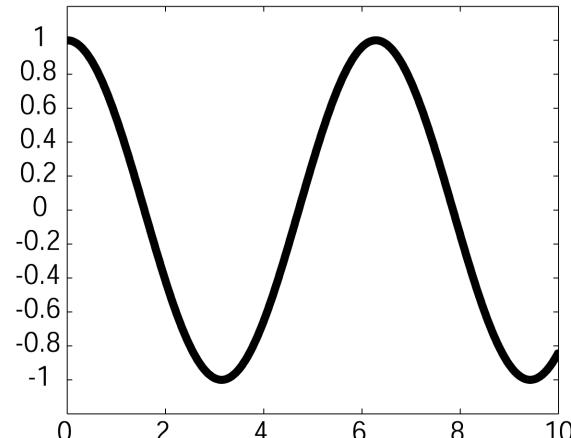
measurement 2

sine (amplitude 1, period 2pi)

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$



cosine (amplitude 1, period 2pi)

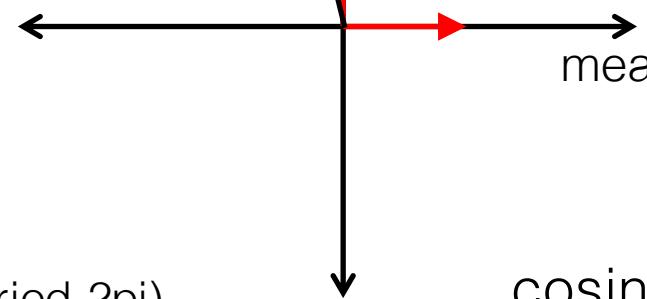
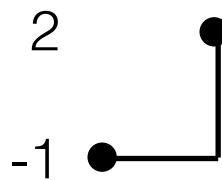


How is this possible?

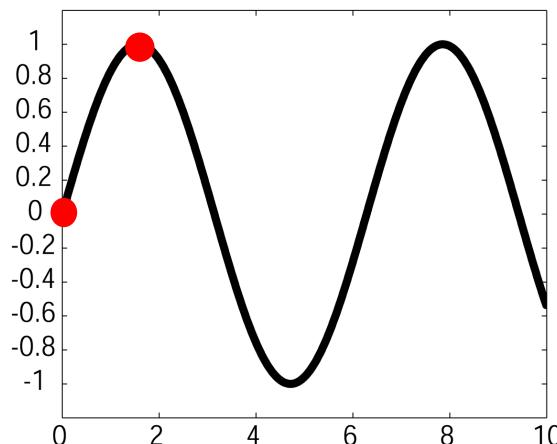
Let's find the "Fourier series" representation of a simple

signal

$$\begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

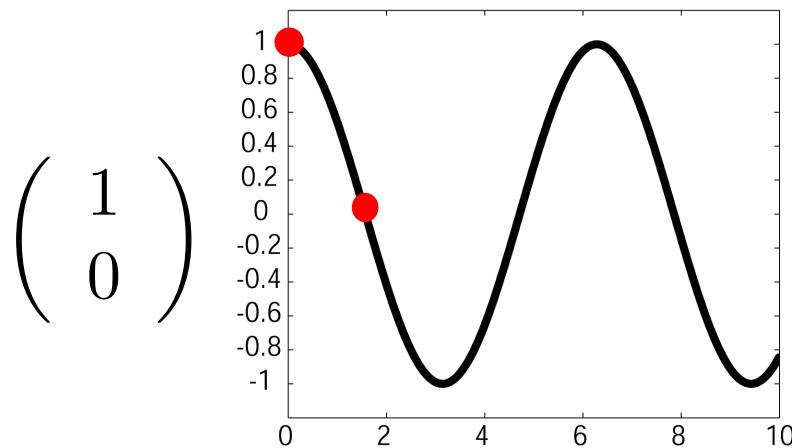


sine (amplitude 1, period 2pi)



$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

cosine (amplitude 1, period 2pi)



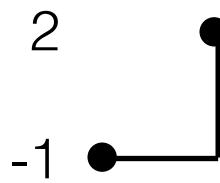
$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

How is this possible?

Let's find the "Fourier series" representation of a simple

signal

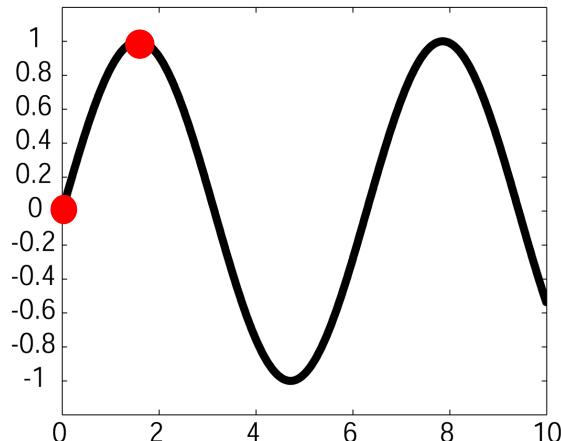
$$\begin{pmatrix} -1 \\ 2 \end{pmatrix}$$



$$2 \sin(x) - \cos(x)$$

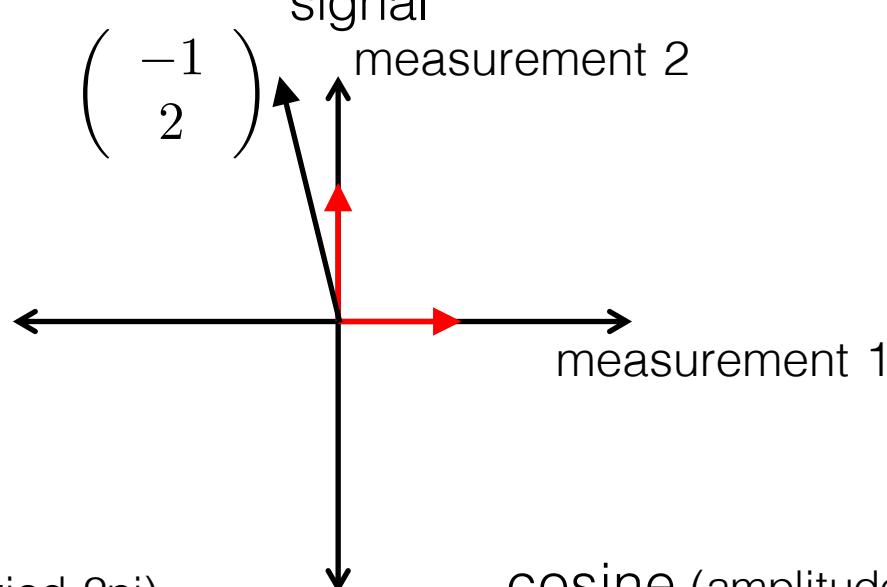
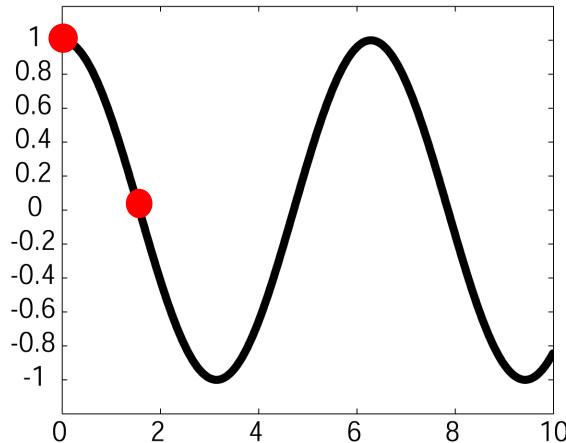
sine (amplitude 1, period 2π)

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$



cosine (amplitude 1, period 2π)

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

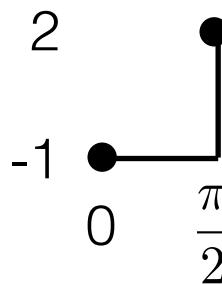


How is this possible?

Let's find the "Fourier series" representation of a simple

signal

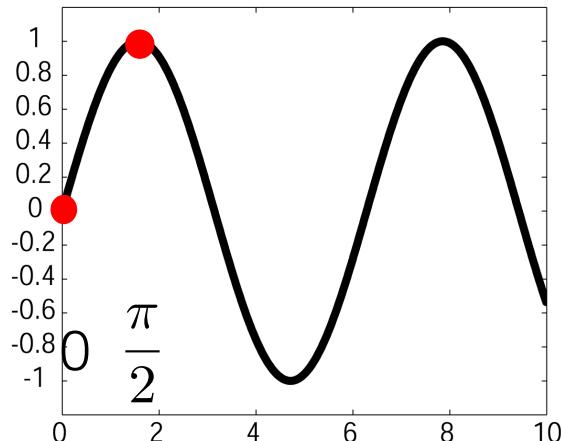
$$\begin{pmatrix} -1 \\ 2 \end{pmatrix}$$



$$2 \sin(x) - \cos(x)$$

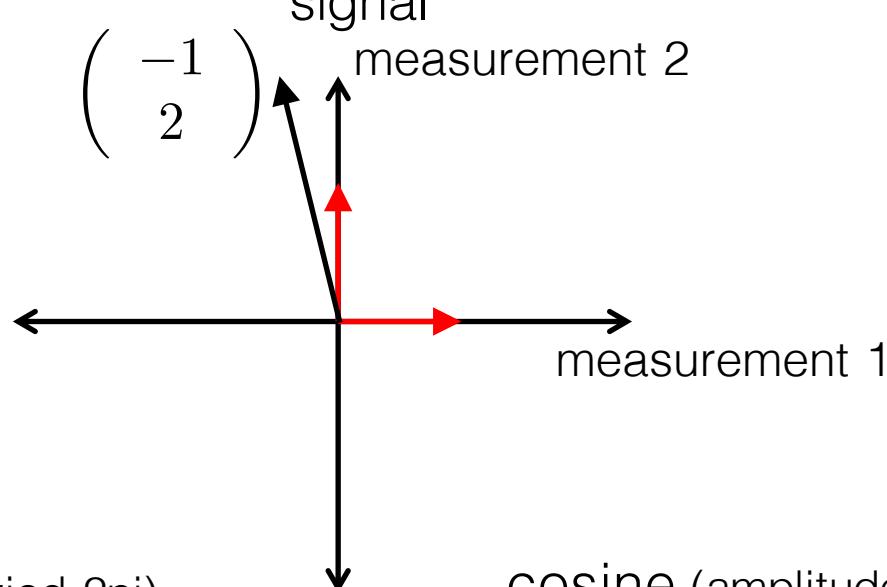
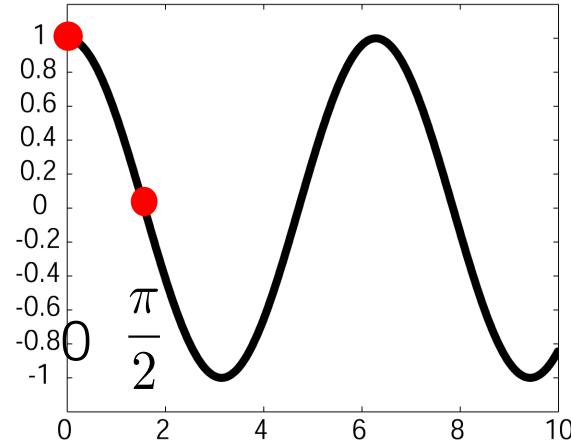
sine (amplitude 1, period 2pi)

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$



cosine (amplitude 1, period 2pi)

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

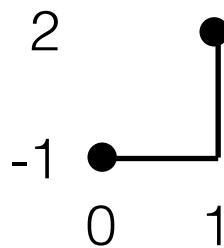


How is this possible?

Let's find the "Fourier series" representation of a simple

signal

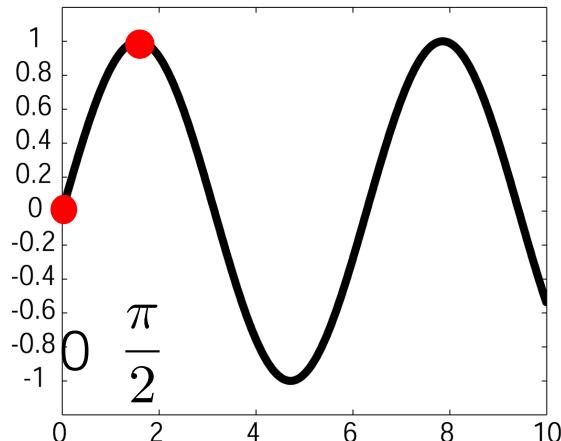
$$\begin{pmatrix} -1 \\ 2 \end{pmatrix}$$



$$2 \sin(x) - \cos(x)$$

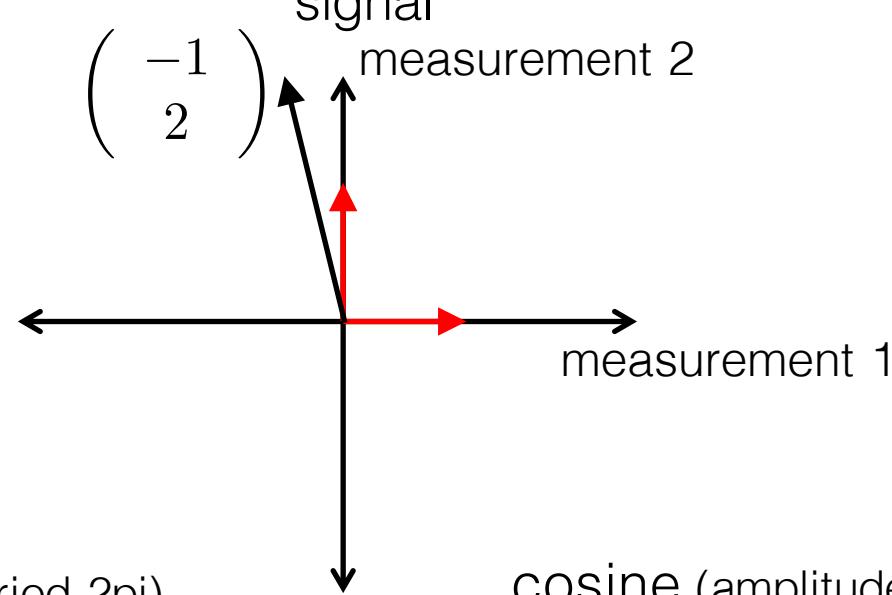
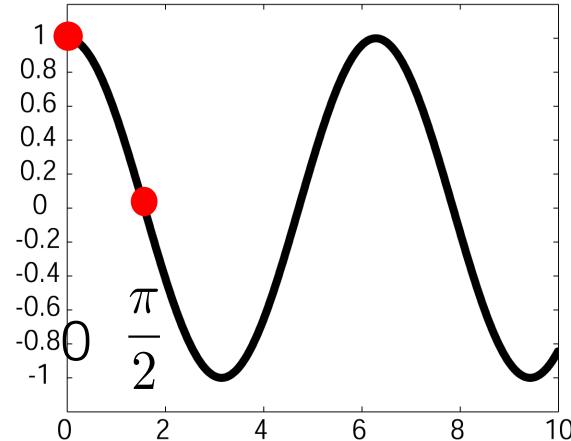
sine (amplitude 1, period 2π)

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$



$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

cosine (amplitude 1, period 2π)



How is this possible?

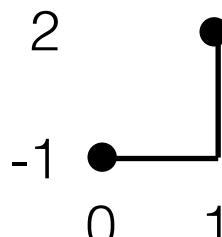
Let's find the "Fourier series" representation of a simple

signal

$$\begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

measurement 1

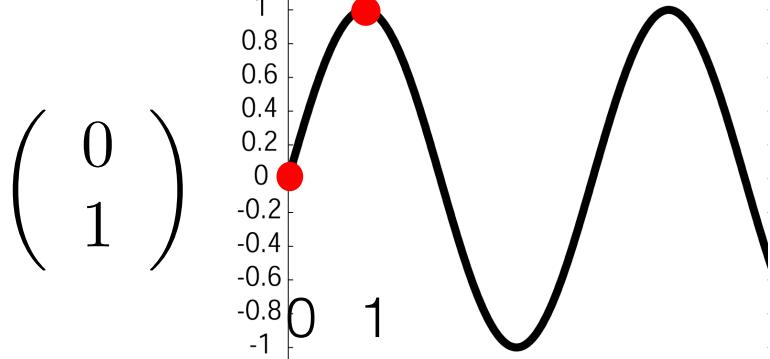
measurement 2



$$2 \sin\left(2\pi \frac{1}{4}x\right) - \cos\left(2\pi \frac{1}{4}x\right)$$

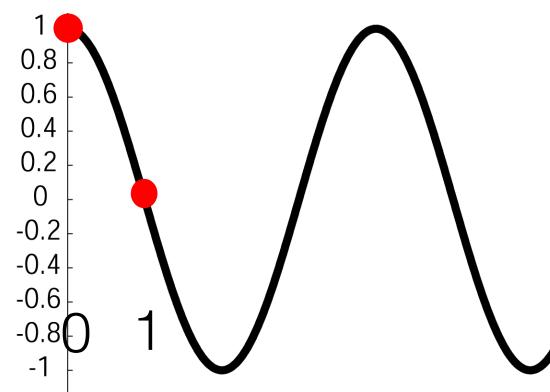
sine (amplitude 1, period 4)

cosine (amplitude 1, period 4)



$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$



How is this possible?

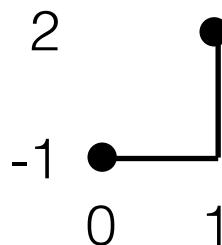
Let's find the "Fourier series" representation of a simple

signal

$$\begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

measurement 1

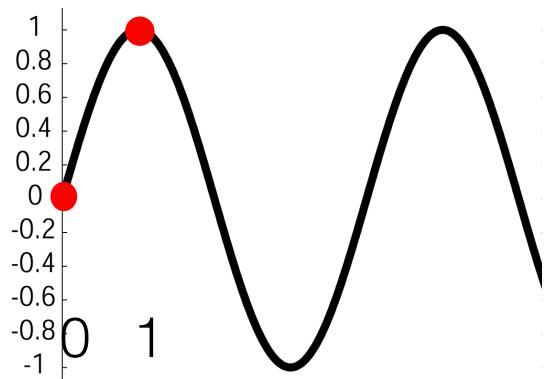
measurement 2



$$2 \sin\left(\frac{\pi}{2}x\right) - \cos\left(\frac{\pi}{2}x\right)$$

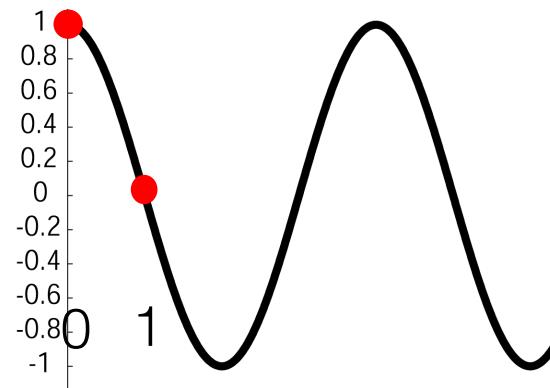
sine (amplitude 1, period 4)

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$



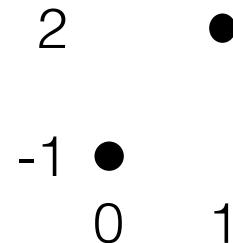
$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

cosine (amplitude 1, period 4)

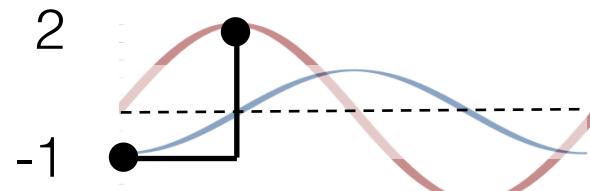


Moving to frequency space

signal measurements
vs
time



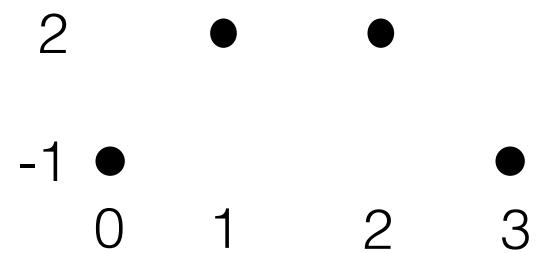
sinusoid coefficients
vs
frequency



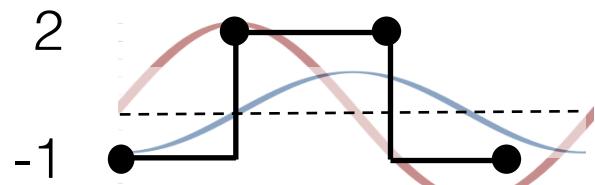
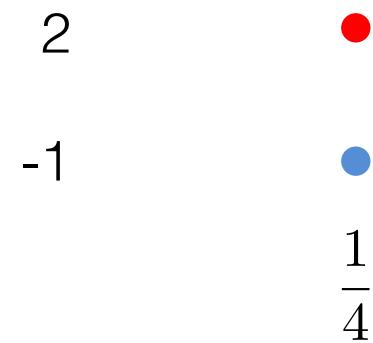
$$-1 \cos\left(\frac{\pi}{2}x\right) + 2 \sin\left(\frac{\pi}{2}x\right)$$

Moving to frequency space

signal measurements
vs
time

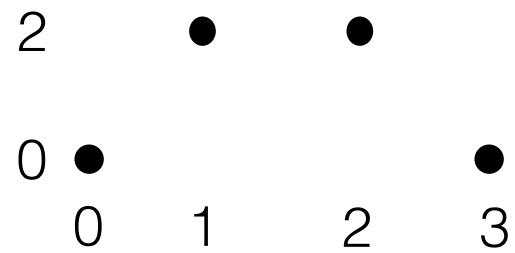


sinusoid coefficients
vs
frequency

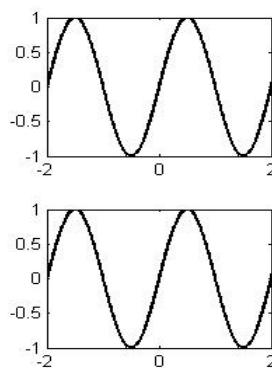
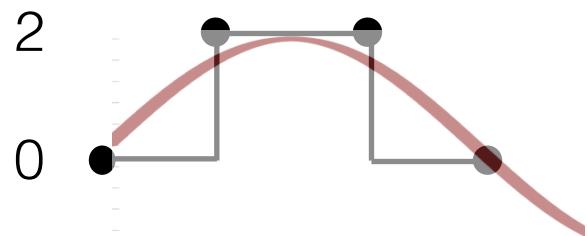
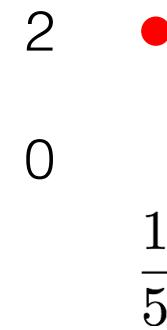


Moving to frequency space

signal measurements
vs
time

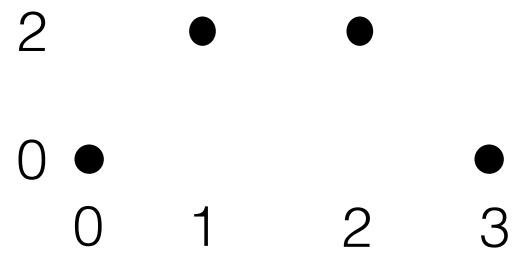


sinusoid coefficients
vs
frequency

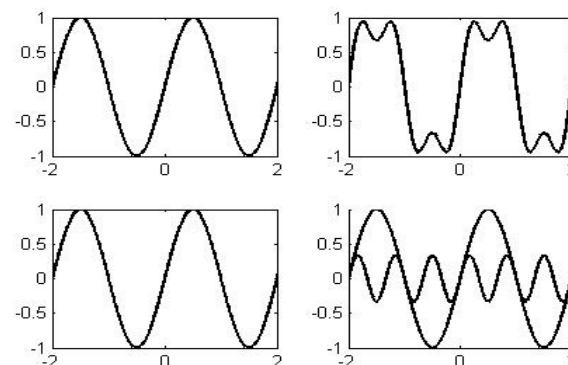
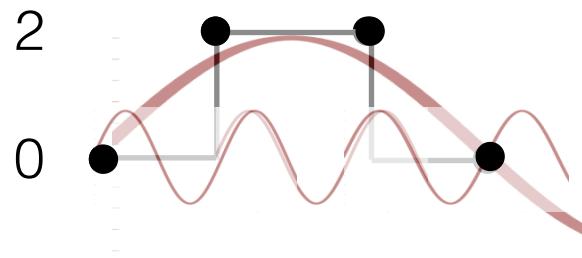
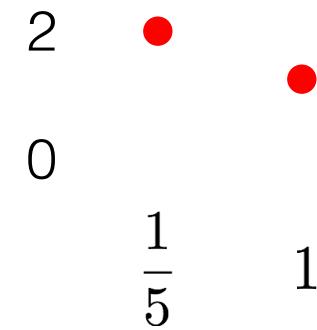


Moving to frequency space

signal measurements
vs
time

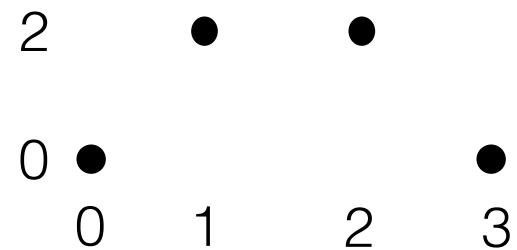


sinusoid coefficients
vs
frequency

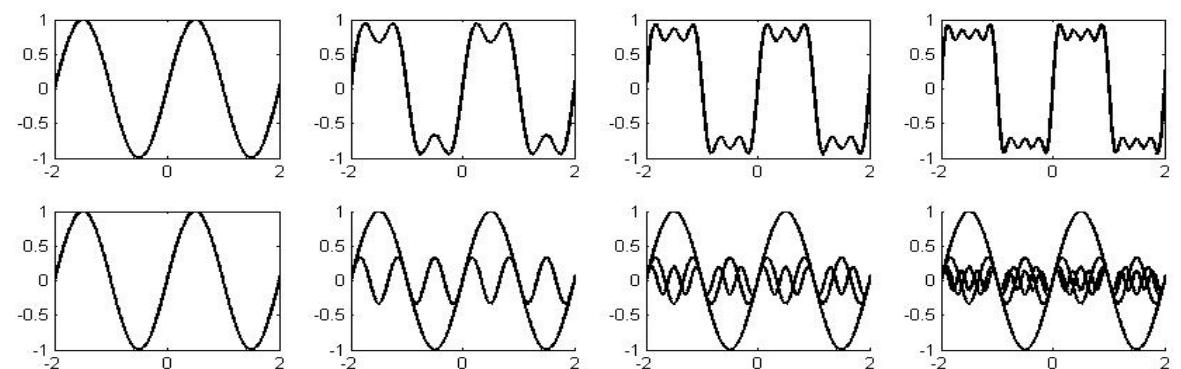
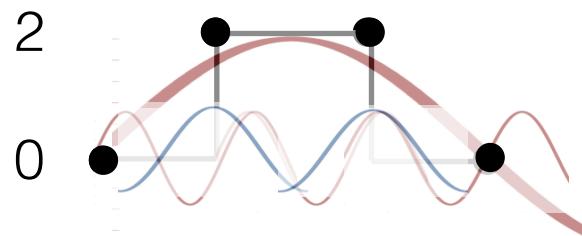
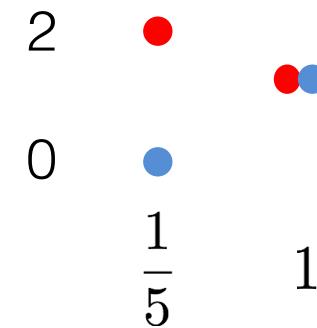


Moving to frequency space

signal measurements
vs
time

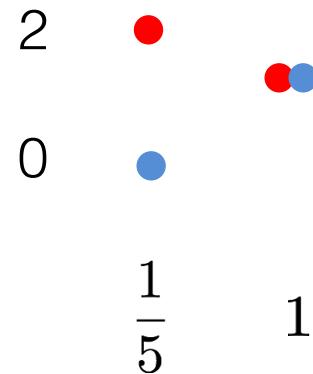


sinusoid coefficients
vs
frequency



Fourier transform

Fourier series

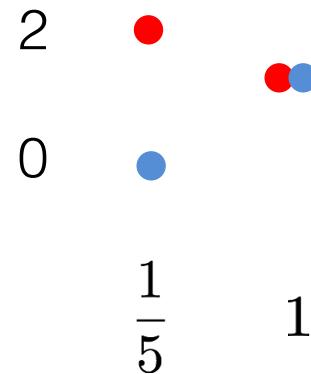


$$0 \cos\left(\frac{2\pi}{5}x\right) + 2 \sin\left(\frac{2\pi}{5}x\right)$$

$$\begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

Fourier transform

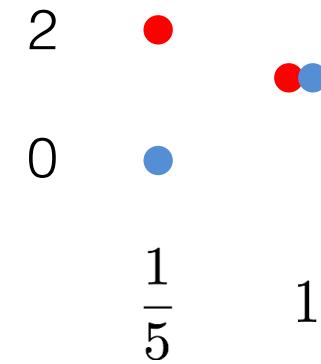
Fourier series



$$0 \cos\left(\frac{2\pi}{5}x\right) + 2 \sin\left(\frac{2\pi}{5}x\right)$$

$$\begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

Fourier transform



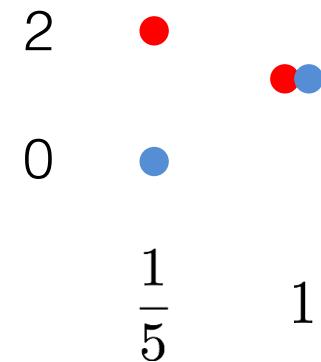
$$0 \cos\left(\frac{2\pi}{5}x\right) - i2 \sin\left(\frac{2\pi}{5}x\right)$$

$$0 - 2i$$

Fourier transform

$$F[\text{signal}]_f = \sum_t^{T-1} \text{signal}(t) \cdot e^{-i2\pi ft}$$

Fourier transform



$$0 \cos\left(\frac{2\pi}{5}x\right) - i2 \sin\left(\frac{2\pi}{5}x\right)$$

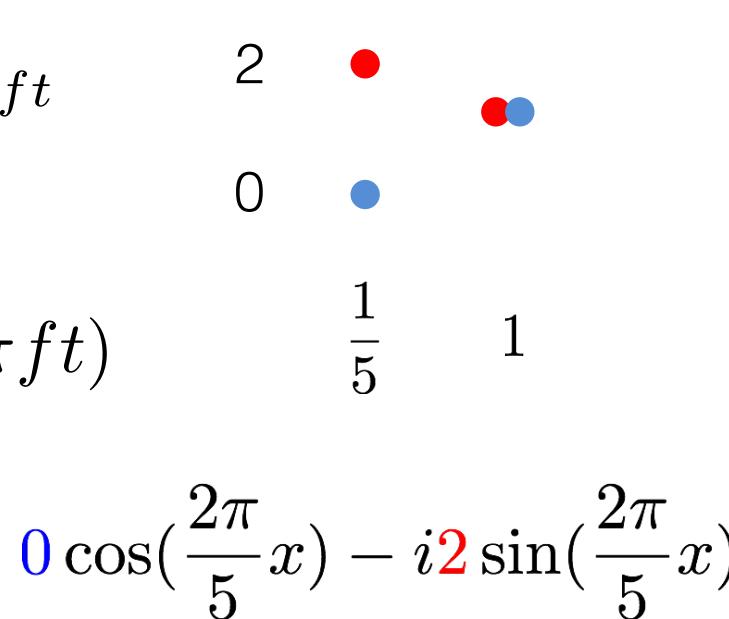
$$0 - 2i$$

MATLAB: `fft(signal)`

Fourier transform

$$F[\text{signal}]_f = \sum_t^{T-1} \text{signal}(t) \cdot e^{-i2\pi ft}$$

$$e^{-i2\pi ft} = \cos(2\pi ft) - i \sin(2\pi ft)$$



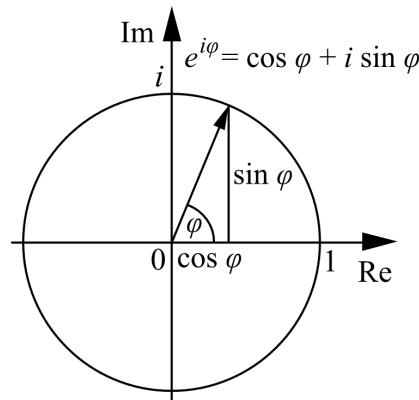
$$0 - 2i$$

MATLAB: `fft(signal)`

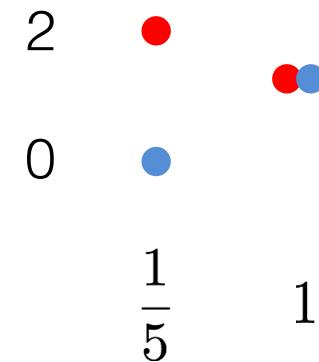
Fourier transform

$$F[\text{signal}]_f = \sum_t^{T-1} \text{signal}(t) \cdot e^{-i2\pi ft}$$

$$e^{-i2\pi ft} = \cos(2\pi ft) - i \sin(2\pi ft)$$



Fourier transform



$$0 \cos\left(\frac{2\pi}{5}x\right) - i 2 \sin\left(\frac{2\pi}{5}x\right)$$

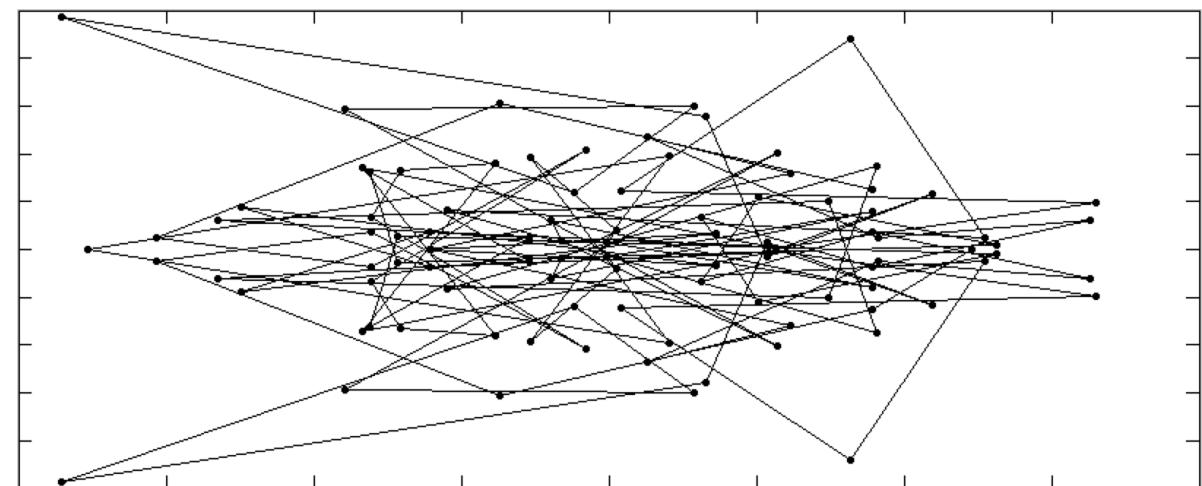
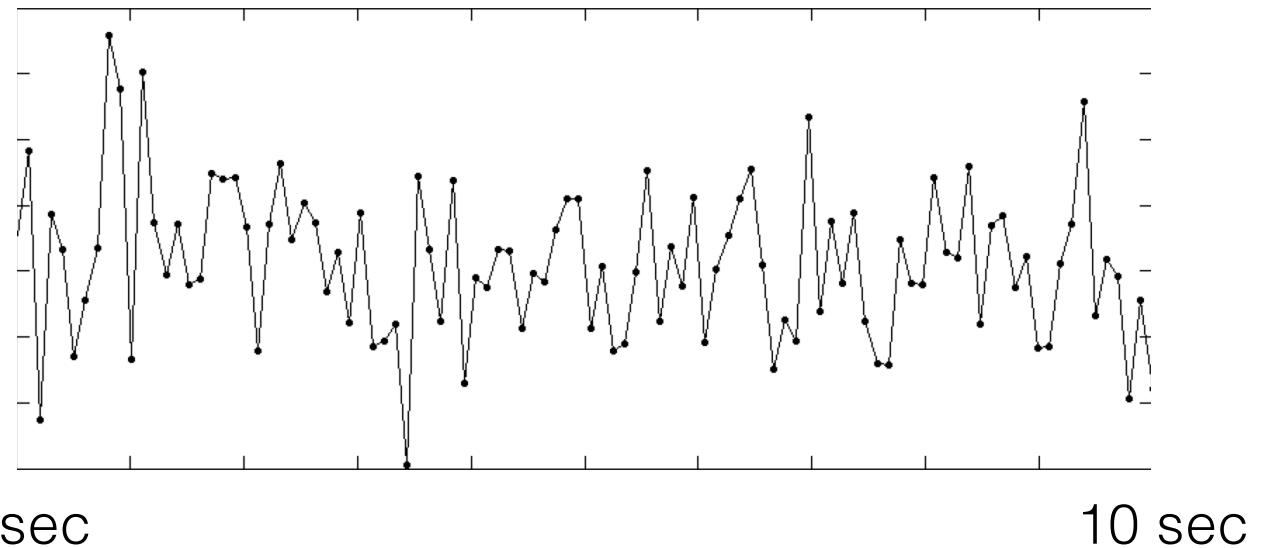
$$0 - 2i$$

MATLAB: `fft(signal)`

Fourier transform example

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```

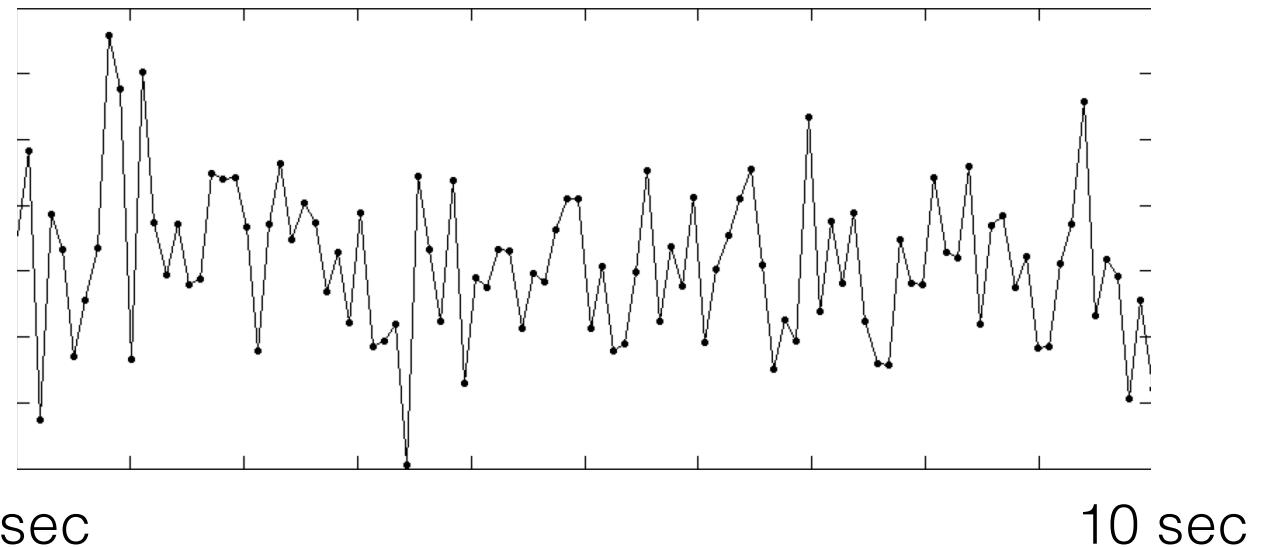


MATLAB: `fft(signal)`

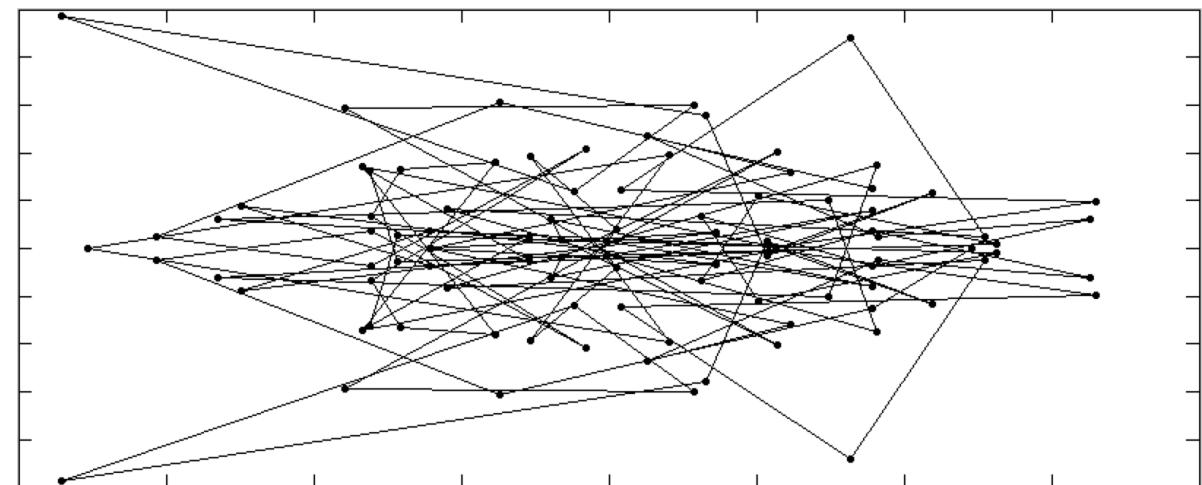
Fourier transform example

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```



```
F      = fft(signal)  
F(1)   = 12.3081  
F(2)   = 8.95 - 6.25i  
F(3)   = -3.71 - 15.2i  
...  
...
```

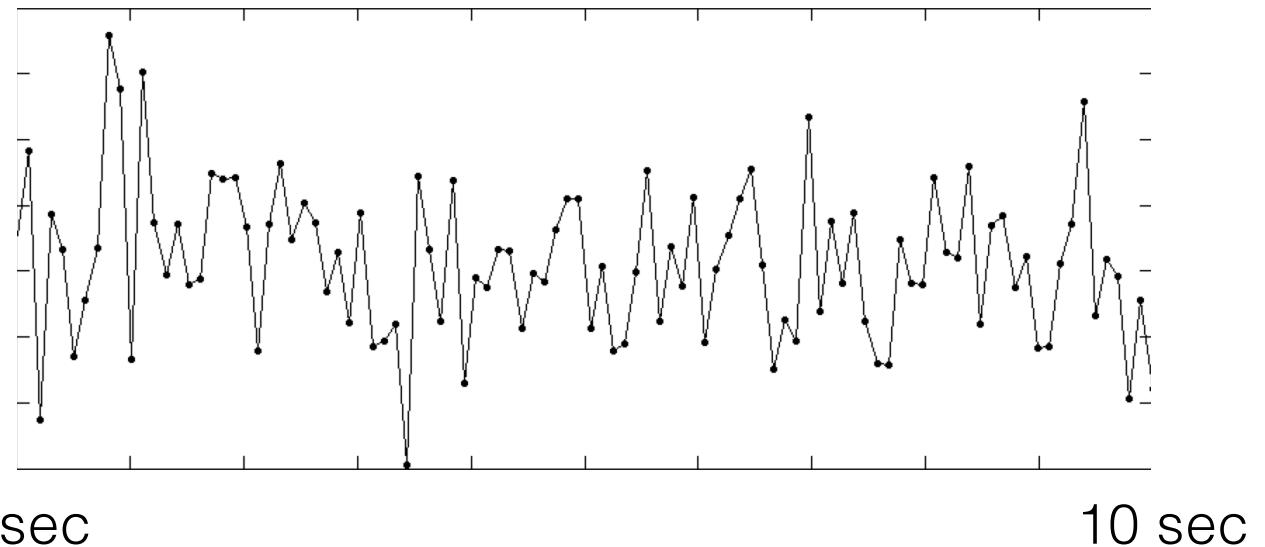


MATLAB: `fft(signal)`

Fourier transform example

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```



```
F = fft(signal)
```

```
F(1) = 12.3081
```

```
F(2) = 8.95 - 6.25i
```

```
F(3) = -3.71 - 15.2i
```

...

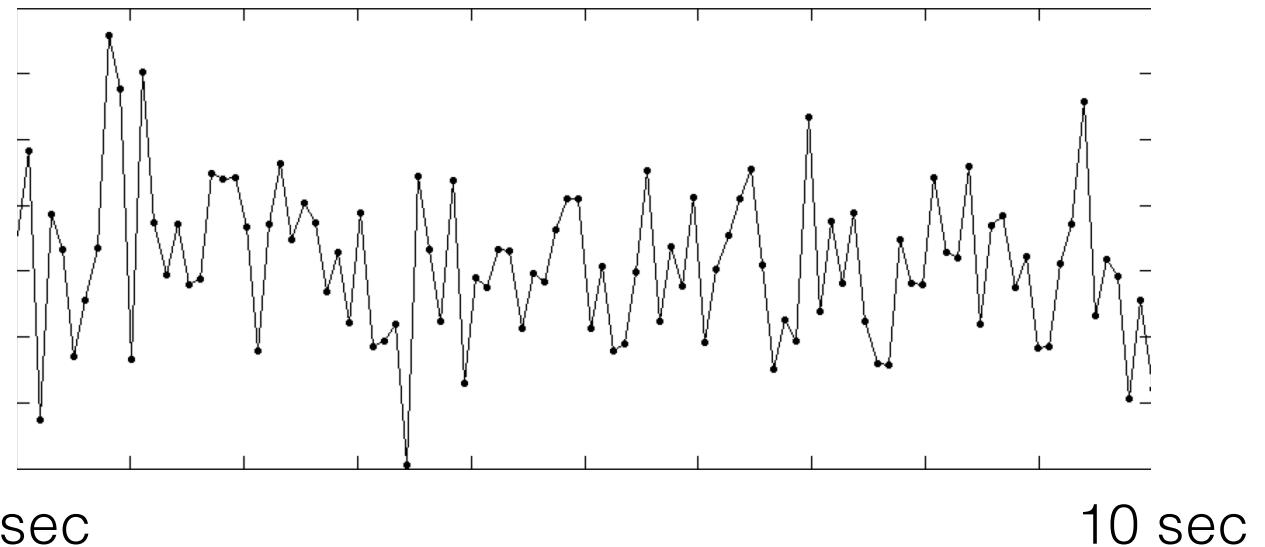
$$F[\text{signal}]_f = \sum_{t=0}^{T-1} \text{signal}(t) \cdot e^{-i2\pi ft}$$

MATLAB: `fft(signal)`

Fourier transform example

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```



```
F = fft(signal)
```

```
F(1) = 12.3081
```

```
F(2) = 8.95 - 6.25i
```

```
F(3) = -3.71 - 15.2i
```

...

typical normalization so
 $F[0 \text{ Hz}] = \text{mean}(\text{signal})$

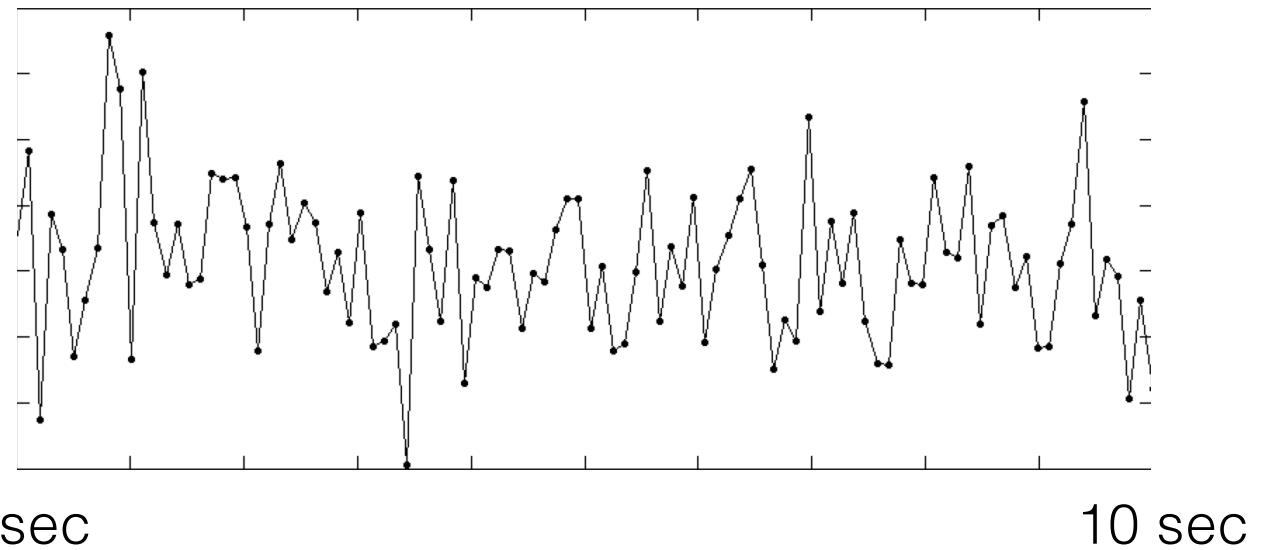
$$F[\text{signal}]_f = \sum_{t=0}^{T-1} \text{signal}(t) \cdot e^{-i2\pi f t}$$

MATLAB: $\overbrace{(1/\text{length}(\text{signal}))}^{\text{normalization}} * \text{fft}(\text{signal})$

Amplitude spectrum example

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```



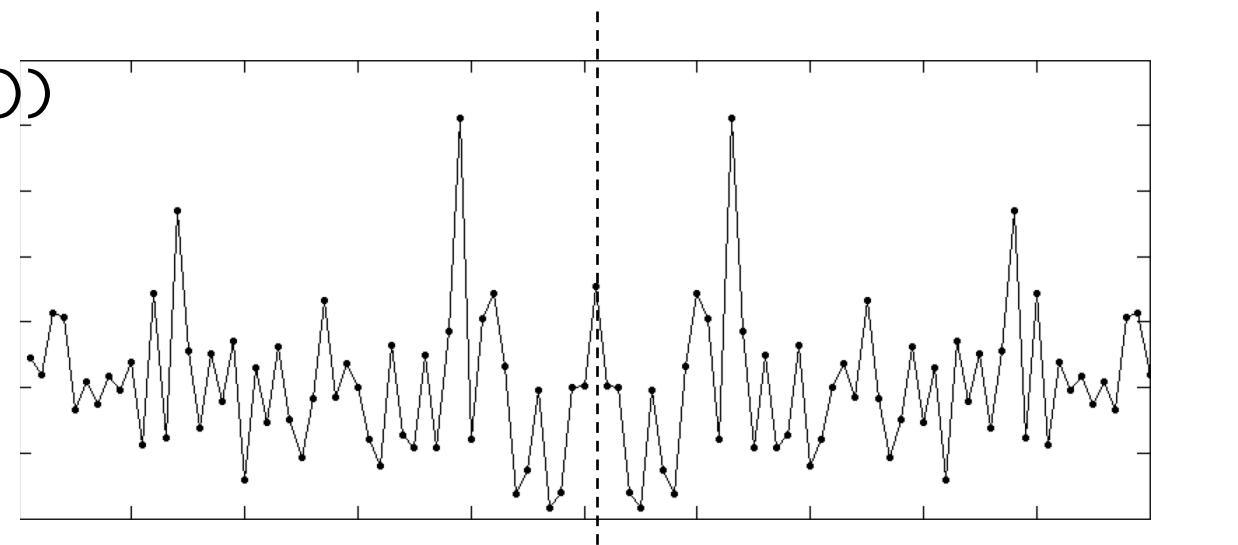
```
F = abs(fft(signal))
```

```
F(1) = 12.3081
```

```
F(2) = 10.9257
```

```
F(3) = 15.6638
```

```
...
```

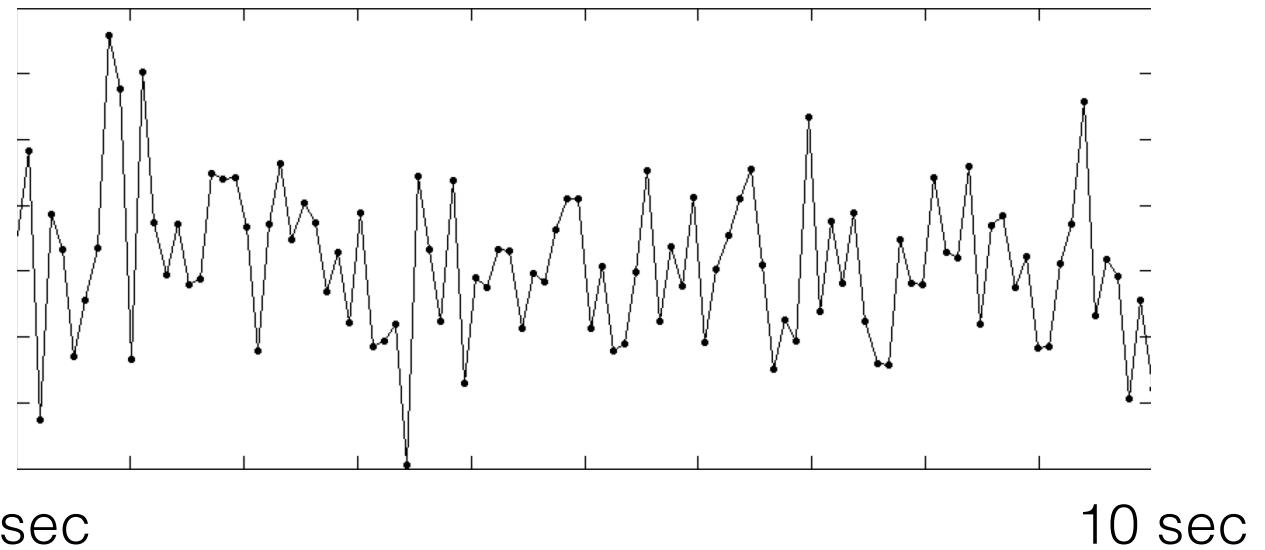


```
MATLAB: abs(fft(signal))
```

Amplitude spectrum example

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```



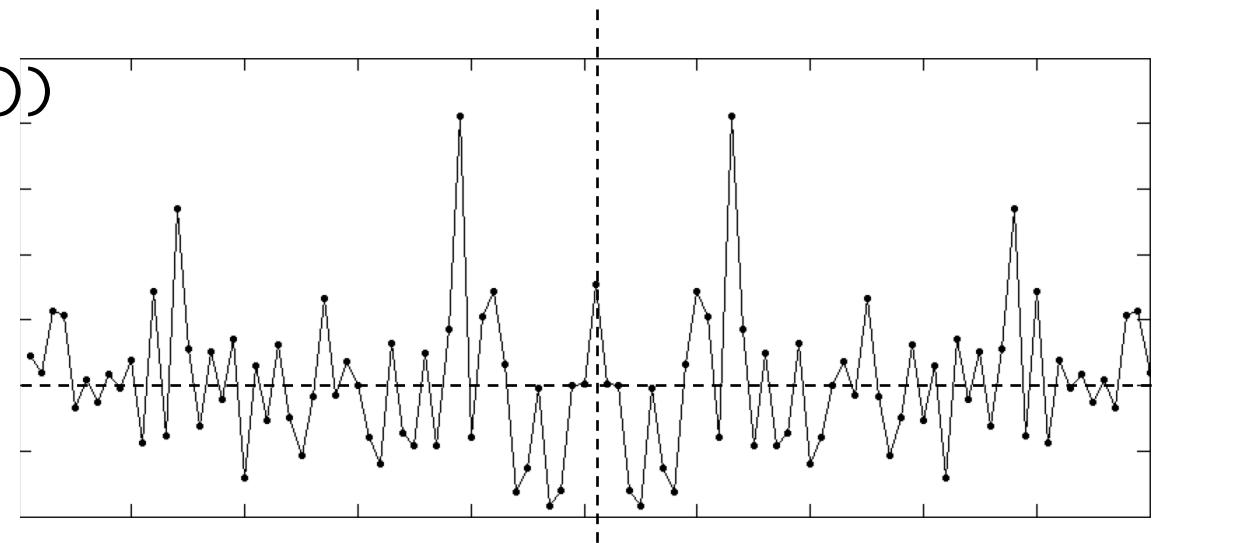
```
F = abs(fft(signal))
```

```
F(1) = 12.3081
```

```
F(2) = 10.9257
```

```
F(3) = 15.6638
```

```
...
```

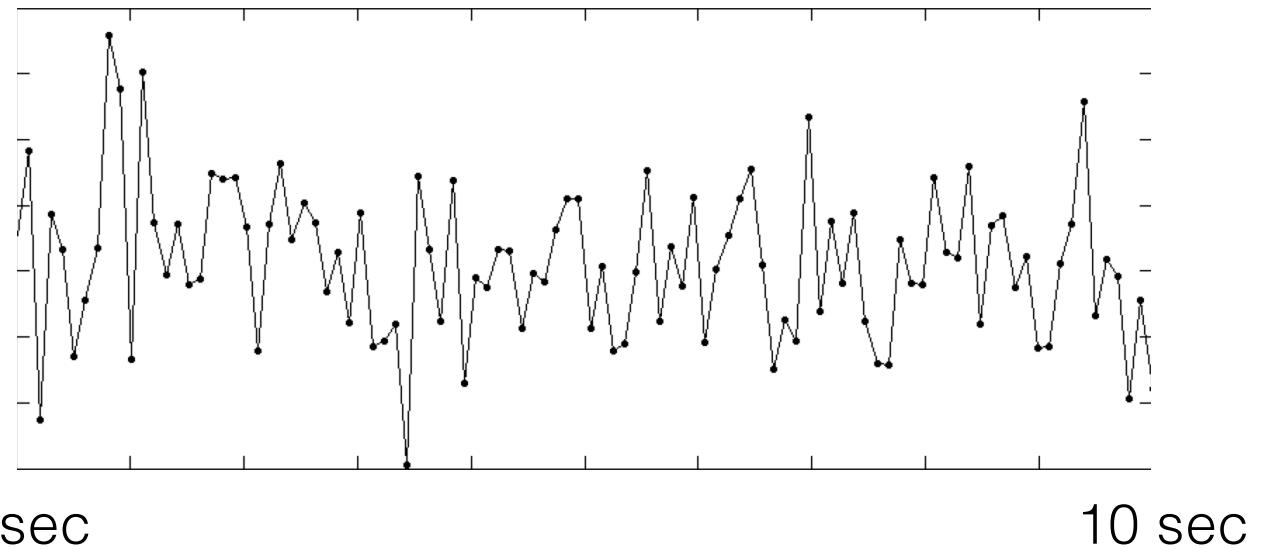


```
MATLAB: abs(fft(signal))
```

Amplitude spectrum example

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```



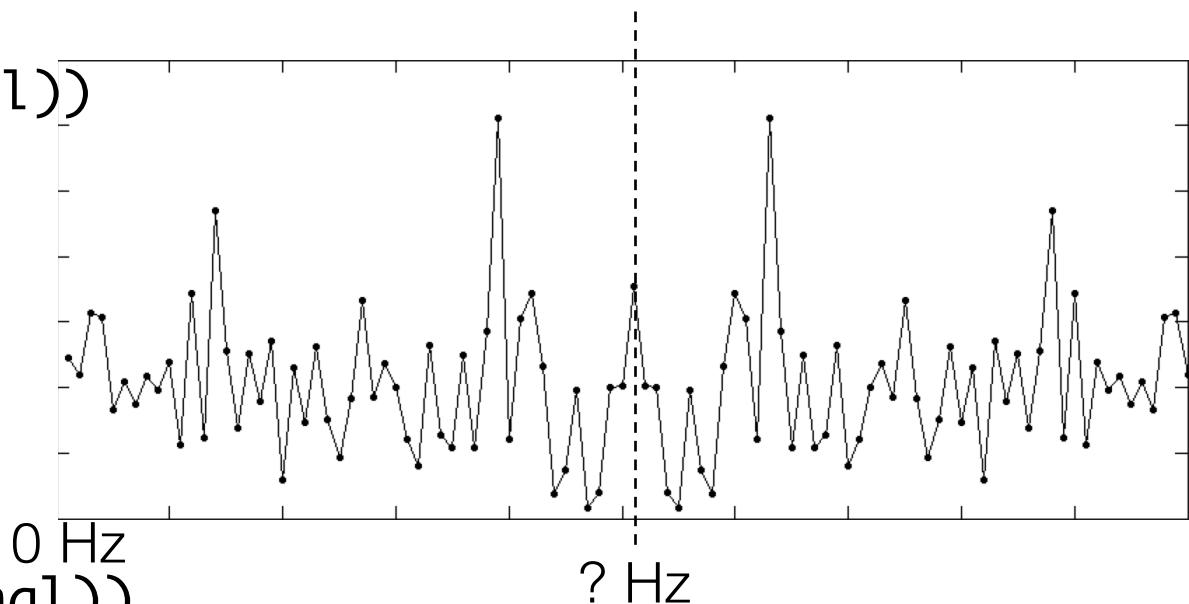
```
F = abs(fft(signal))
```

```
F(1) = 12.3081
```

```
F(2) = 10.9257
```

```
F(3) = 15.6638
```

```
...
```

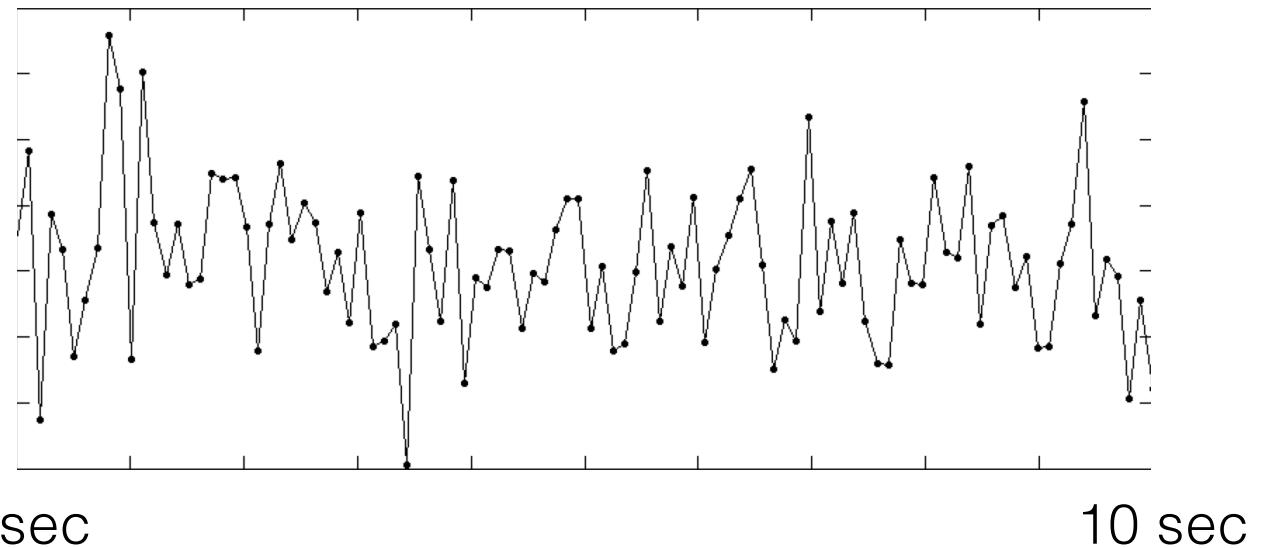


```
MATLAB: abs(fft(signal))
```

Amplitude spectrum example

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```



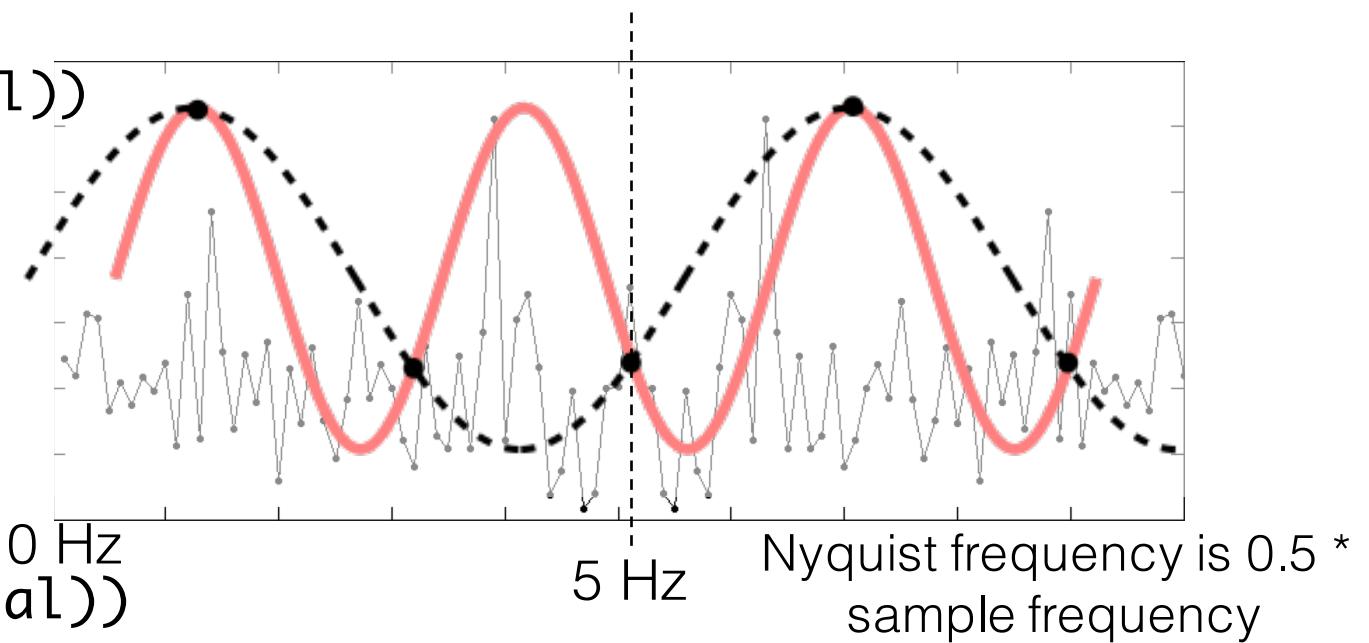
F = abs(fft(signal))

F(1) = 12.3081

F(2) = 10.9257

F(3) = 15.6638

...

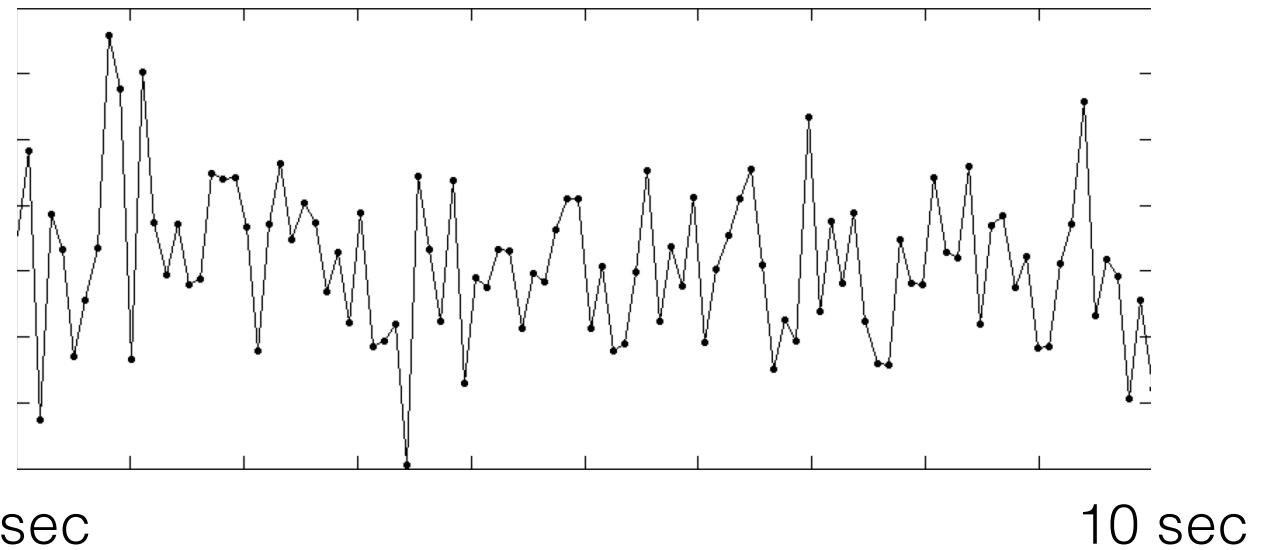


MATLAB: abs(fft(signal))

Amplitude spectrum example

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```



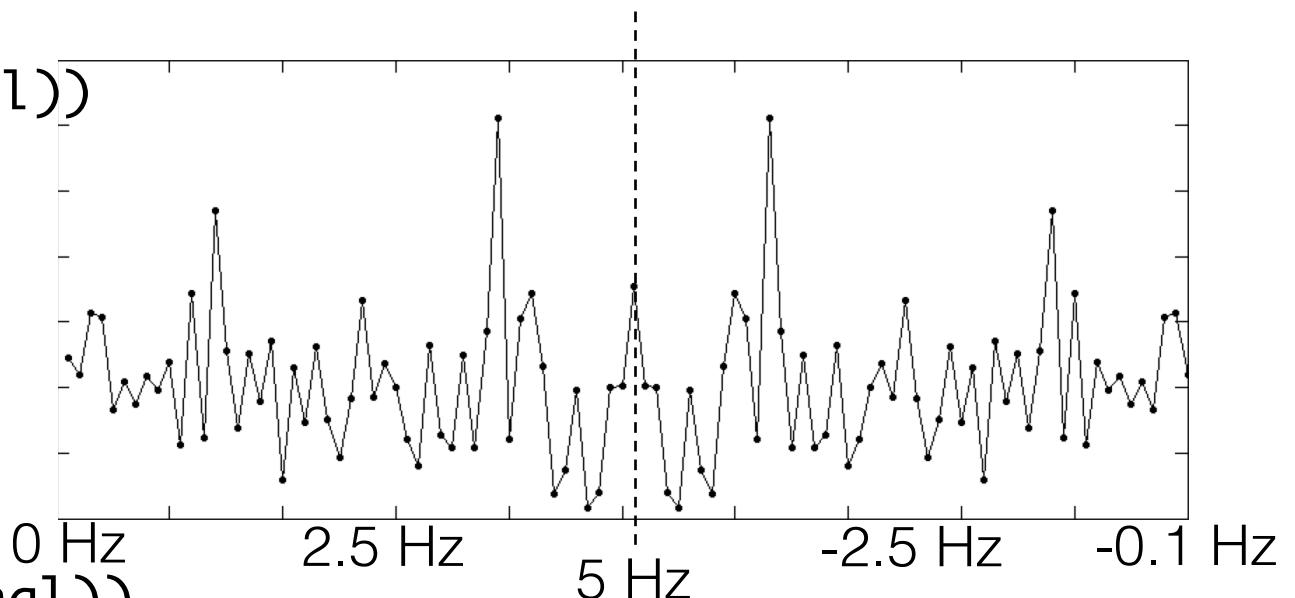
```
F = abs(fft(signal))
```

```
F(1) = 12.3081
```

```
F(2) = 10.9257
```

```
F(3) = 15.6638
```

```
...
```

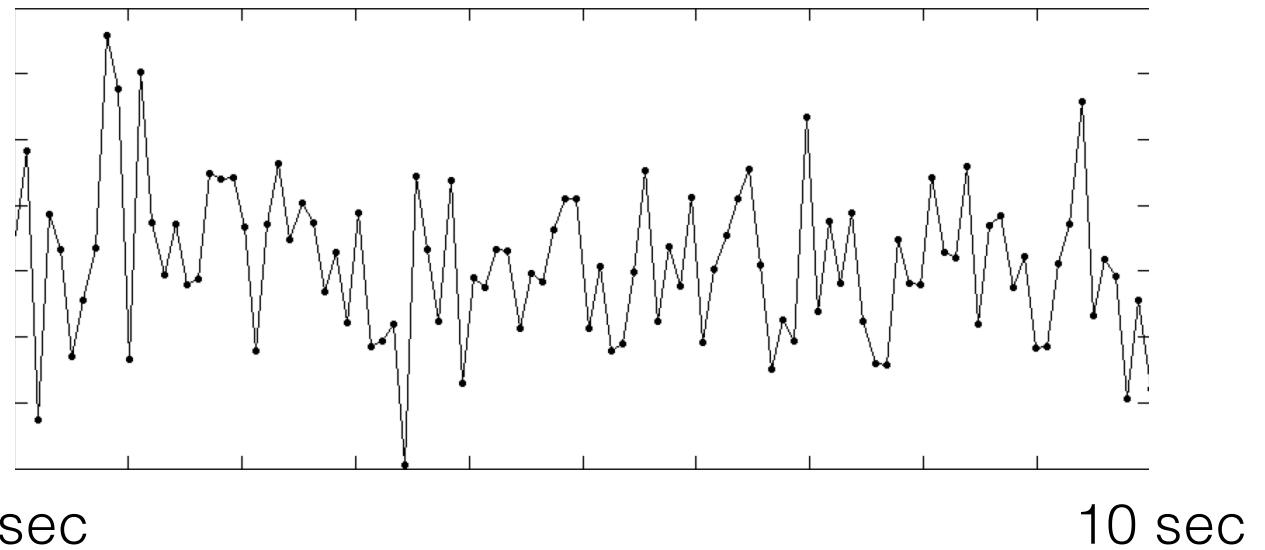


```
MATLAB: abs(fft(signal))
```

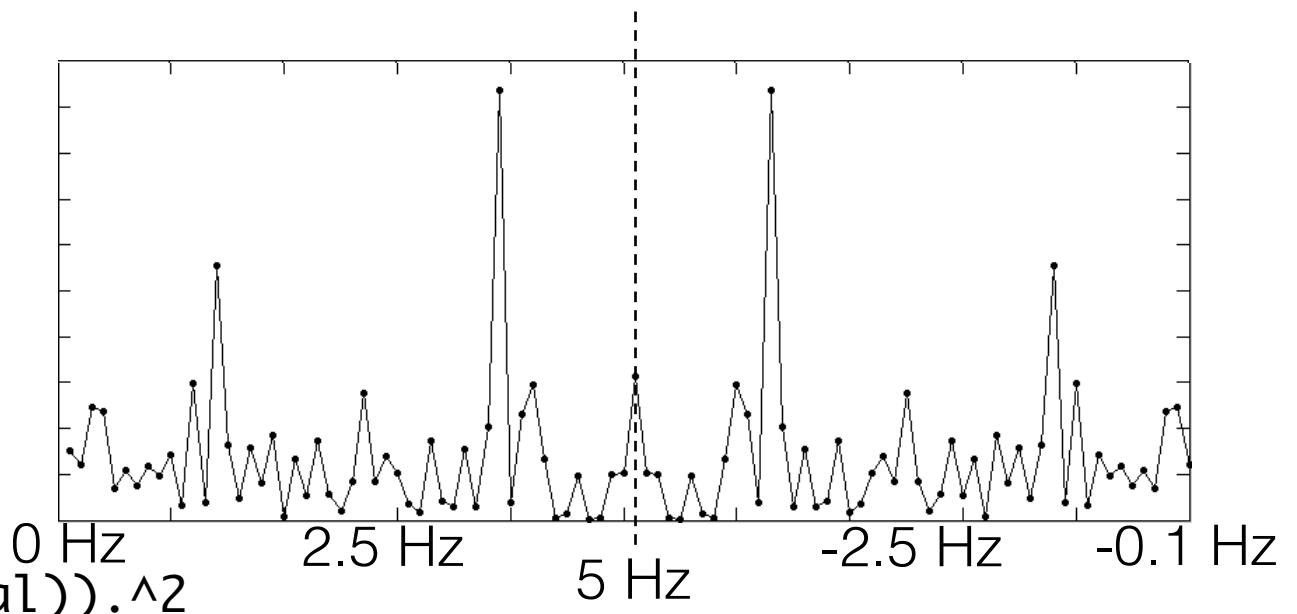
Power spectrum example

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```



power spectrum is just
the squared amplitude
spectrum

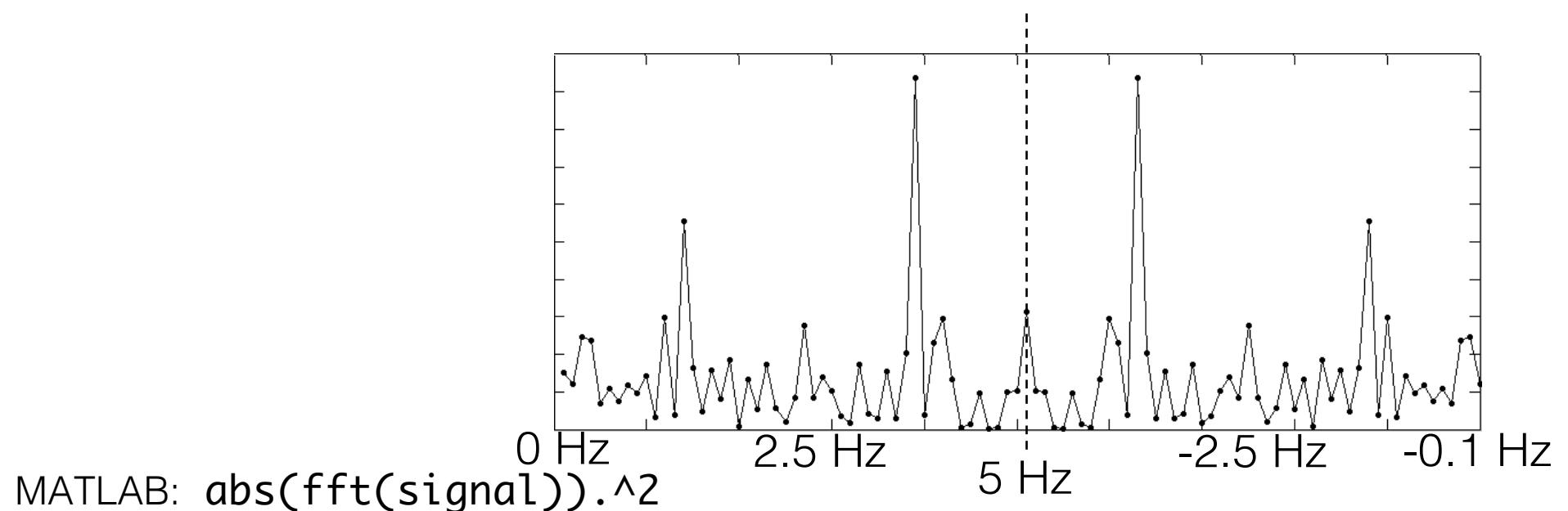
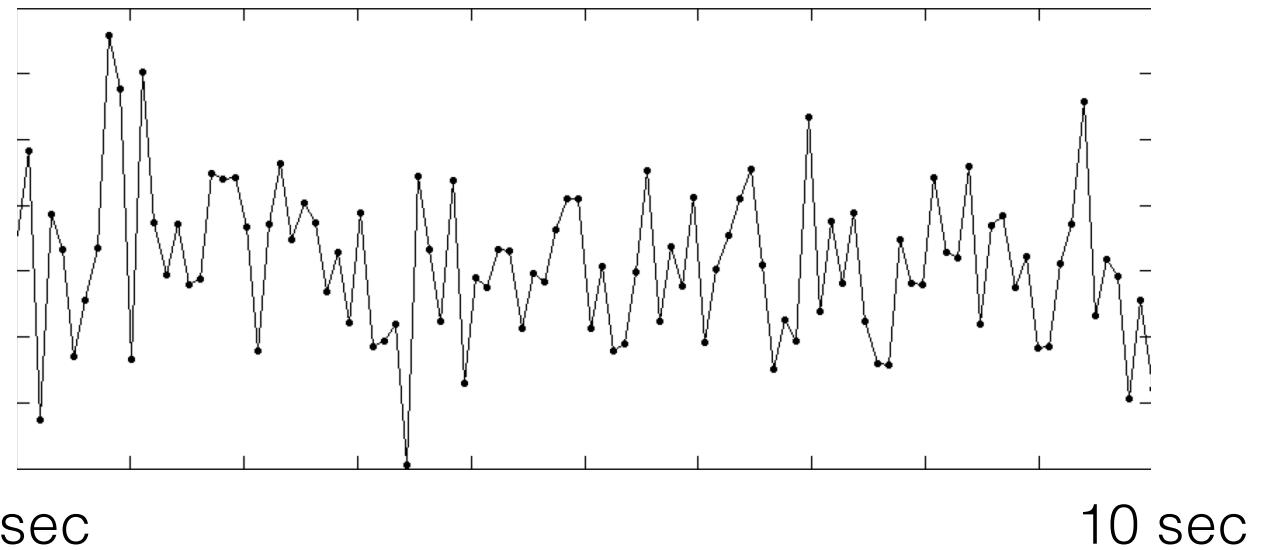


MATLAB: `abs(fft(signal)).^2`

What part do I actually care about?

random white noise
sampled at 10 Hz

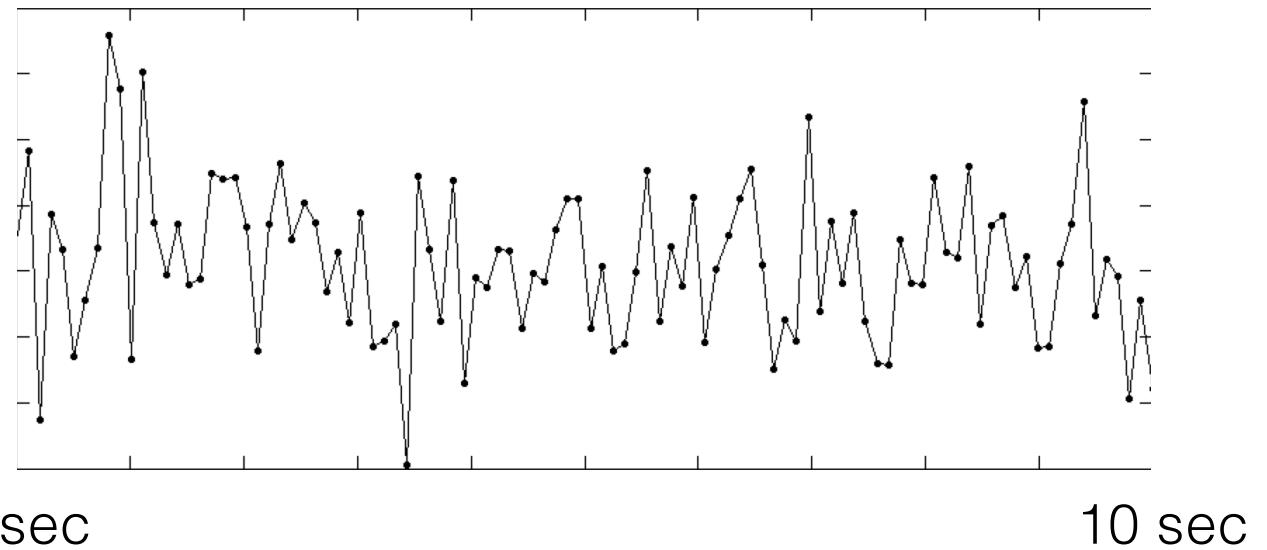
```
randn(100,1)  
linspace(0,10,100)
```



What part do I actually care about?

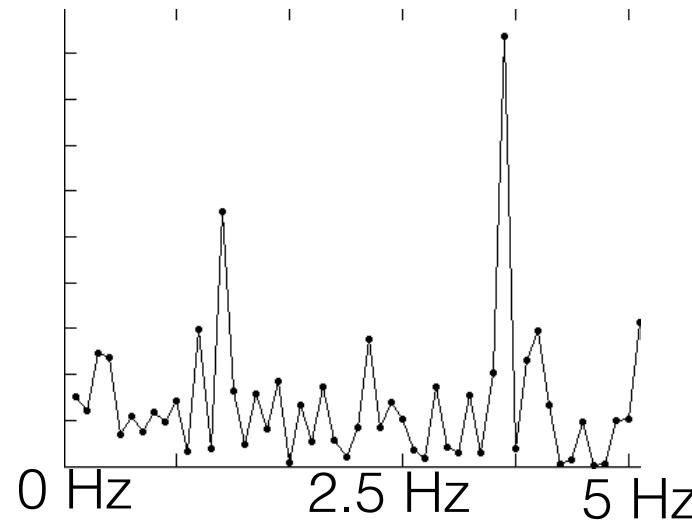
random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```



MATLAB (even length signal):

```
Ps = abs(fft(signal)).^2;  
Ps = Ps(1:length(signal)/2+1)
```



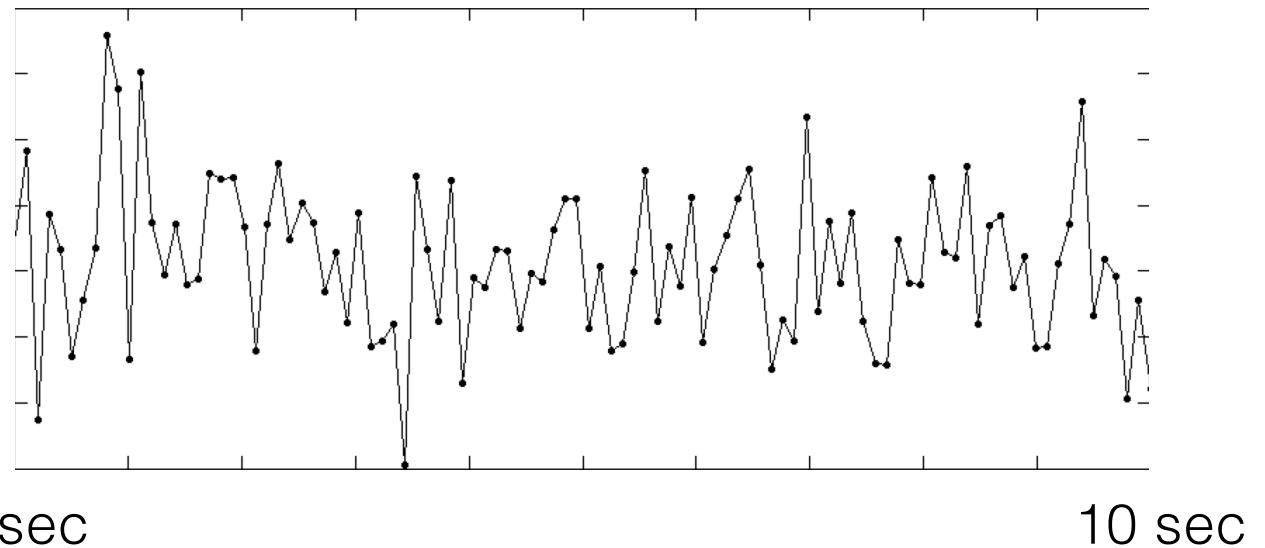
MATLAB (odd length signal):

```
Ps = abs(fft(signal)).^2;  
Ps = Ps(1:(length(signal)-1)/2+1)
```

How to generate x-axis?

random white noise
sampled at 10 Hz

```
randn(100,1)  
linspace(0,10,100)
```

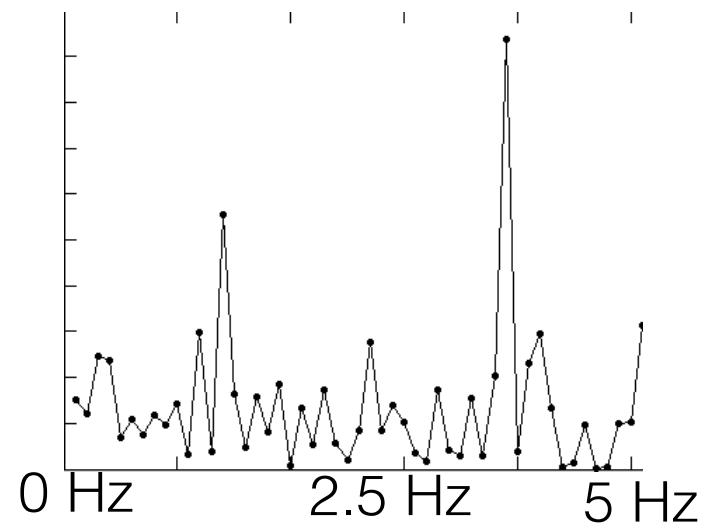


```
%% In general %%  
%% Fs is sample rate %%  
freqs = linspace(0, Fs/2, length(Ps))
```

```
%% Our example %%  
freqs = linspace(0, 5, 51)
```

MATLAB (even length signal):

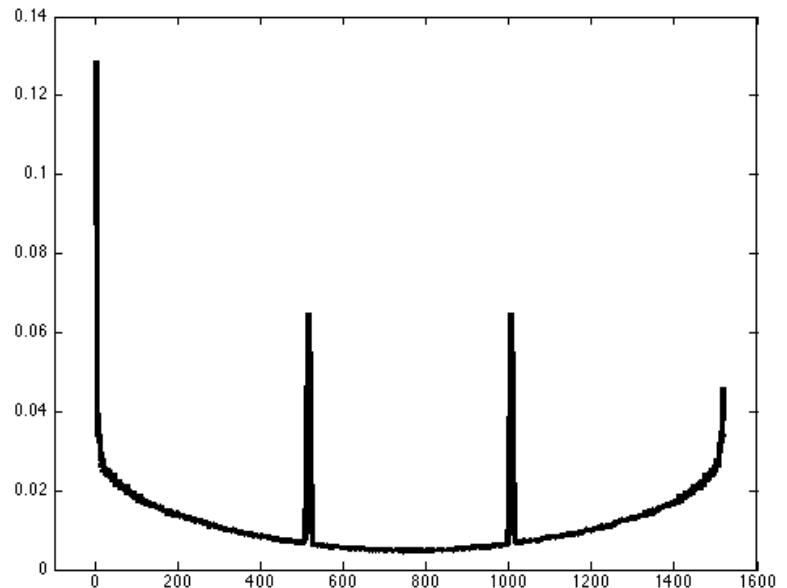
```
Ps = abs(fft(signal)).^2;  
Ps = Ps(1:length(signal)/2+1)
```



Example: De-noising images



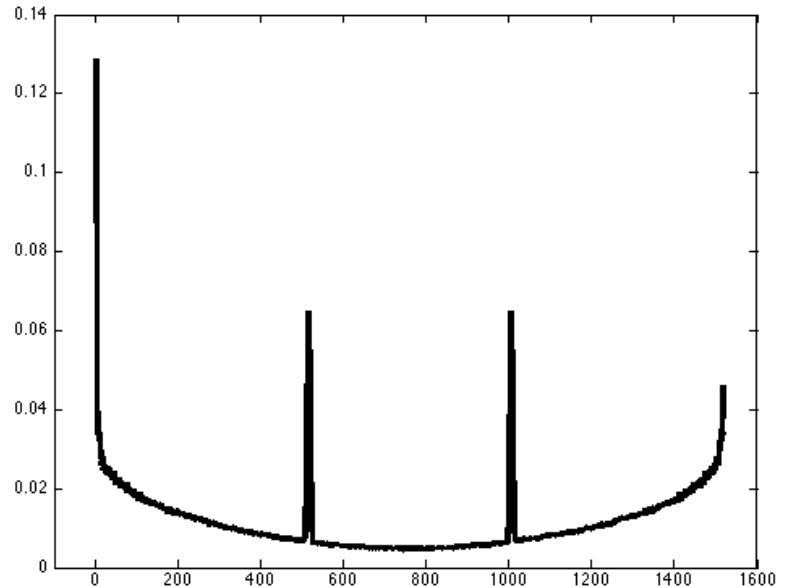
Example: De-noising images



MATLAB (odd length signal):

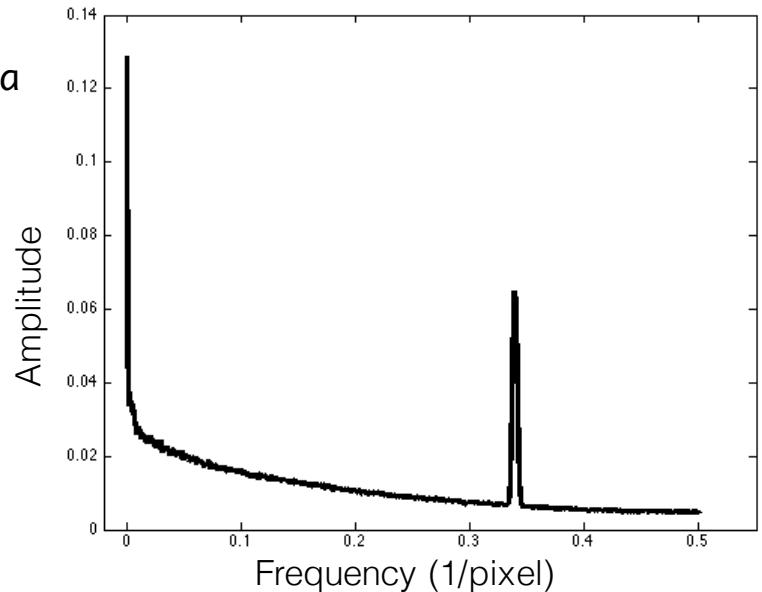
```
As_2d = (1/length(image(:)))*abs(fft2(image));  
As_1d = mean(As_2d, 1); % average all horizontal spectra
```

Example: De-noising images

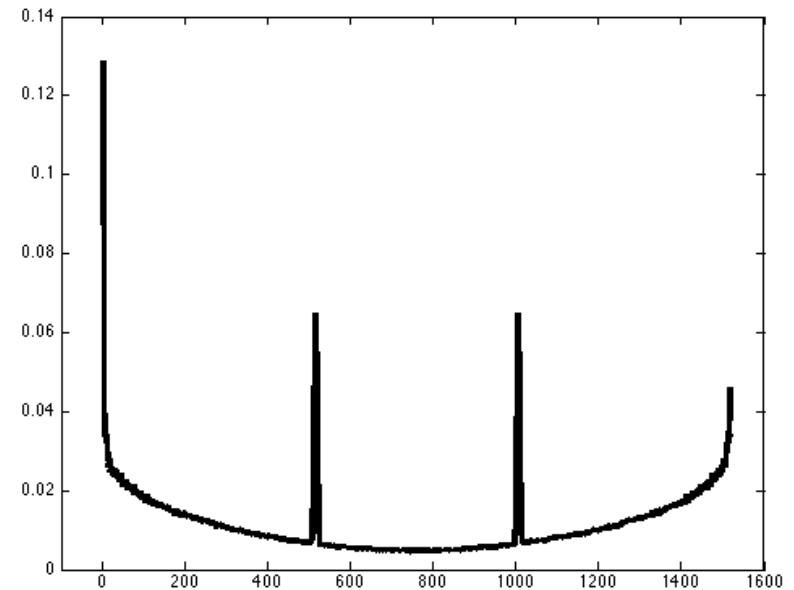


MATLAB (odd length signal):

```
As_2d = (1/length(image(:)))*abs(fft2(image));
As_1d = mean(As_2d, 1) % average all horizontal spectra
As    = As_1d(1:(length(As_1d)-1)/2+1)
freqs = linspace(0, 1/2, length(As))
```

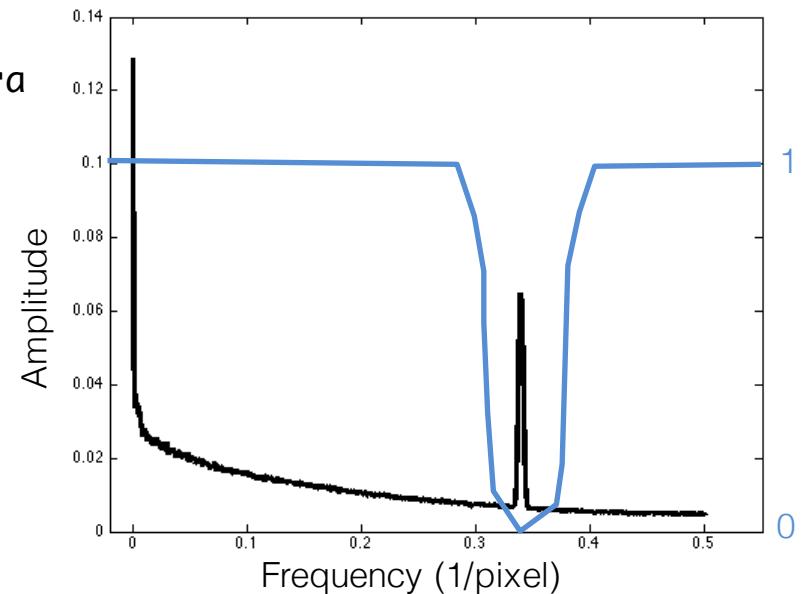


Example: notch filter

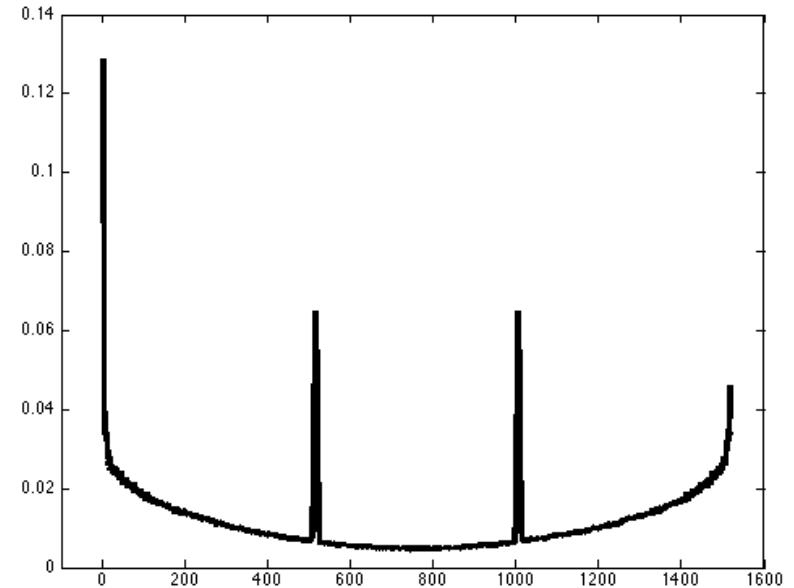


MATLAB (odd length signal):

```
As_2d = (1/length(image(:)))*abs(fft2(image));
As_1d = mean(As_2d, 1) % average all horizontal spectra
As    = As_1d(1:(length(As_1d)-1)/2+1)
freqs = linspace(0, 1/2, length(As))
```

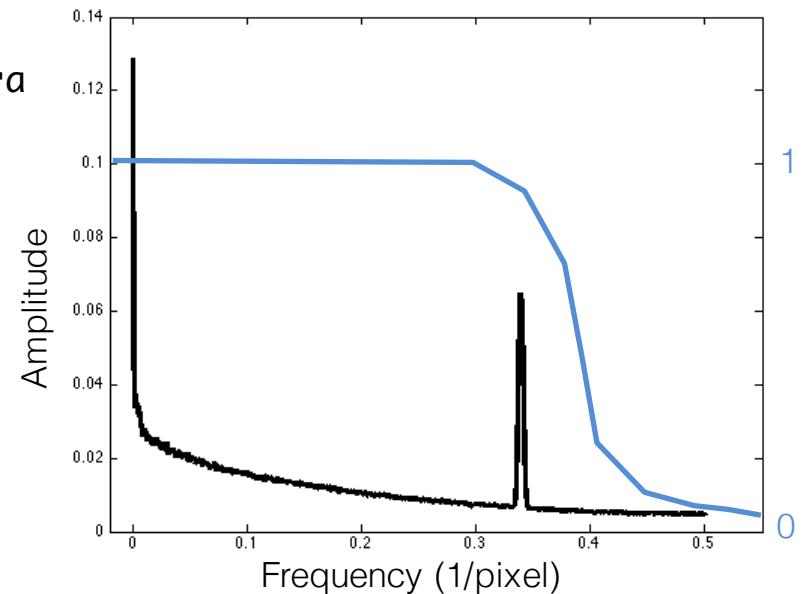


Example: low-pass filter

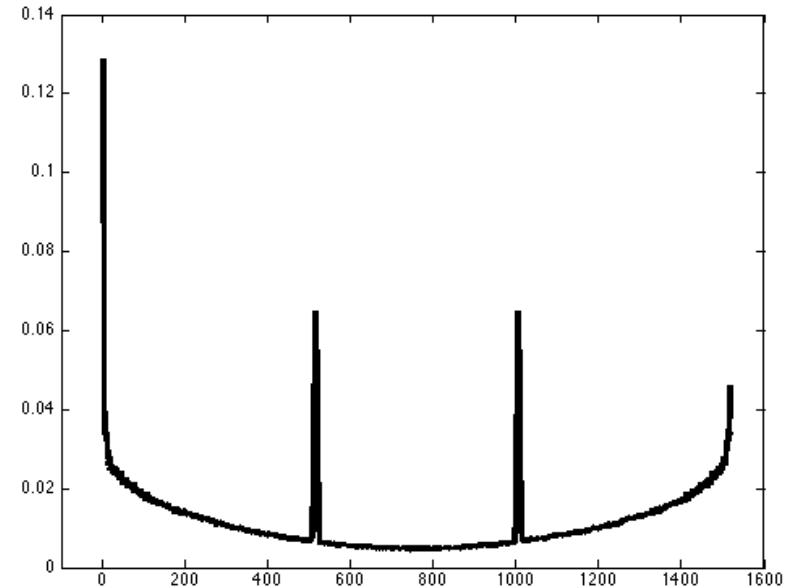


MATLAB (odd length signal):

```
As_2d = (1/length(image(:)))*abs(fft2(image));  
As_1d = mean(As_2d, 1) % average all horizontal spectra  
As    = As_1d(1:(length(As_1d)-1)/2+1)  
freqs = linspace(0, 1/2, length(As))
```

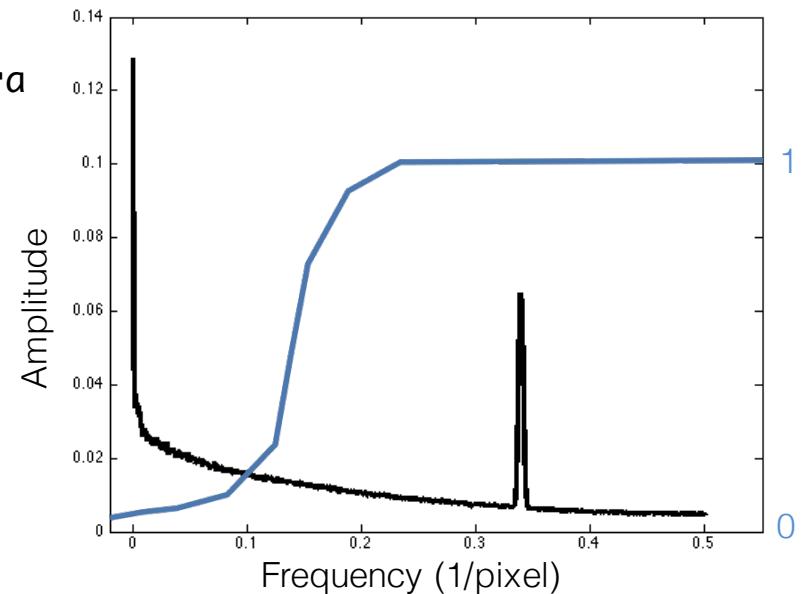


Example: high-pass filter

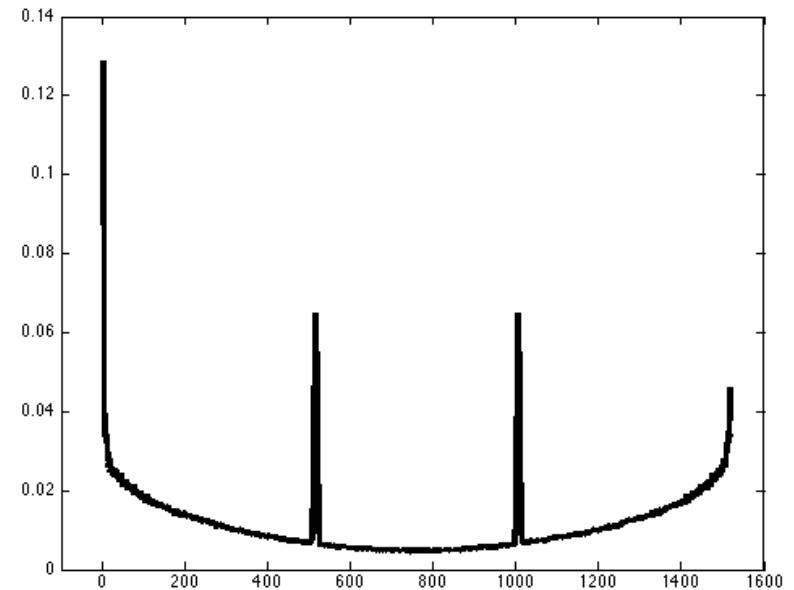


MATLAB (odd length signal):

```
As_2d = (1/length(image(:)))*abs(fft2(image));  
As_1d = mean(As_2d, 1) % average all horizontal spectra  
As    = As_1d(1:(length(As_1d)-1)/2+1)  
freqs = linspace(0, 1/2, length(As))
```

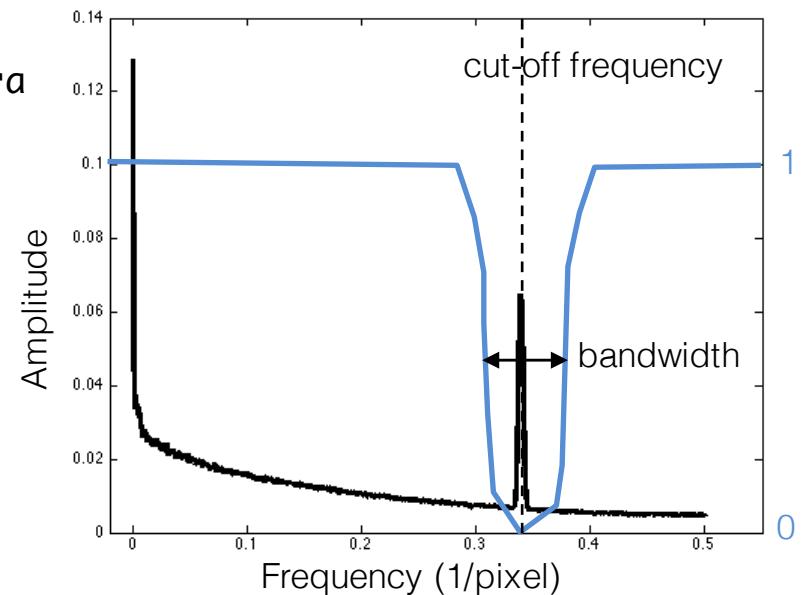


Example: notch filter



MATLAB (odd length signal):

```
As_2d = (1/length(image(:)))*abs(fft2(image));
As_1d = mean(As_2d, 1) % average all horizontal spectra
As    = As_1d(1:(length(As_1d)-1)/2+1)
freqs = linspace(0, 1/2, length(As))
```



Example: notch filter

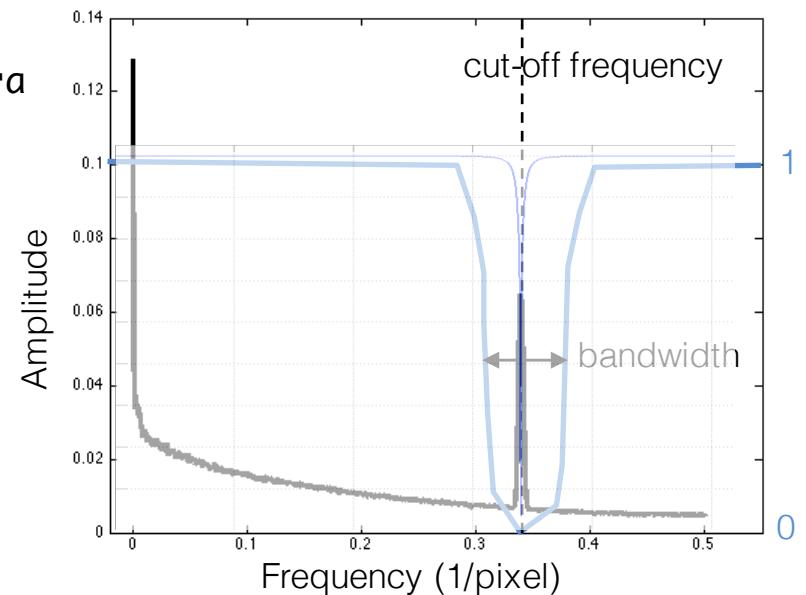
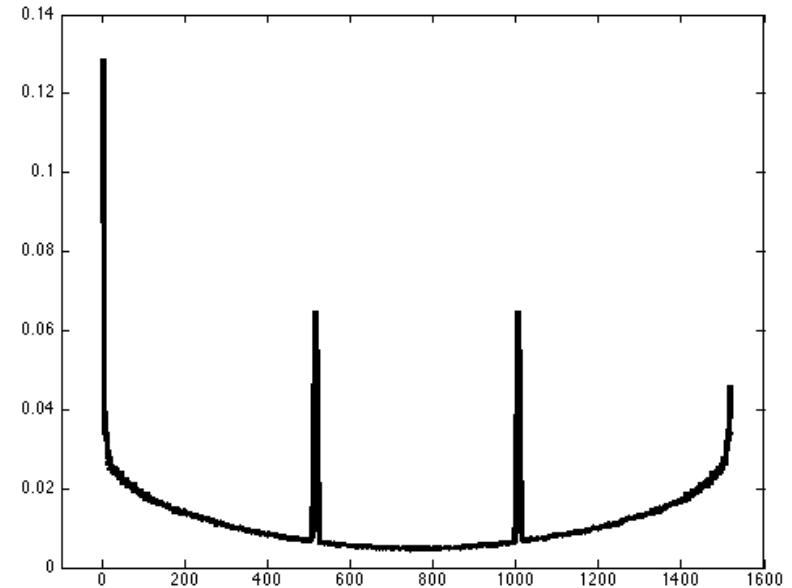


MATLAB (odd length signal):

```
As_2d = (1/length(image(:)))*abs(fft2(image));
As_1d = mean(As_2d, 1) % average all horizontal spectra
As    = As_1d(1:(length(As_1d)-1)/2+1)
freqs = linspace(0, 1/2, length(As))
```

MATLAB (notch filter design):

```
cutoff      = 0.34
bandwidth   = .05
[num, den]  = iirnotch(cutoff, bandwidth)
denoised_im = filter(num, den, image)
```

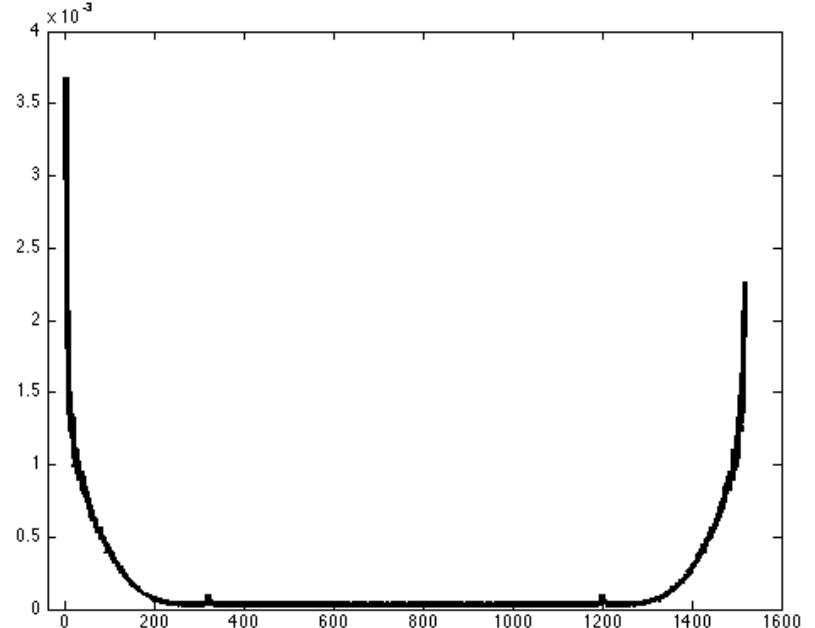




Example: De-noising images



Example: De-noising images

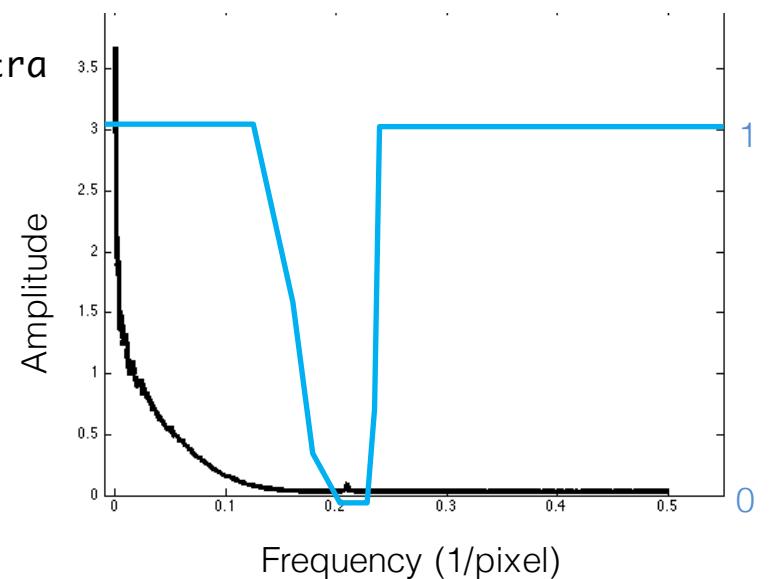


MATLAB (odd length signal):

```
As_2d = (1/length(image(:)))*abs(fft2(image));
As_1d = mean(As_2d, 1); % average all horizontal spectra
As    = As_1d(1:(length(As_1d)-1)/2+1);
freqs = linspace(0, 1/2, length(As));
```

MATLAB (notch filter design):

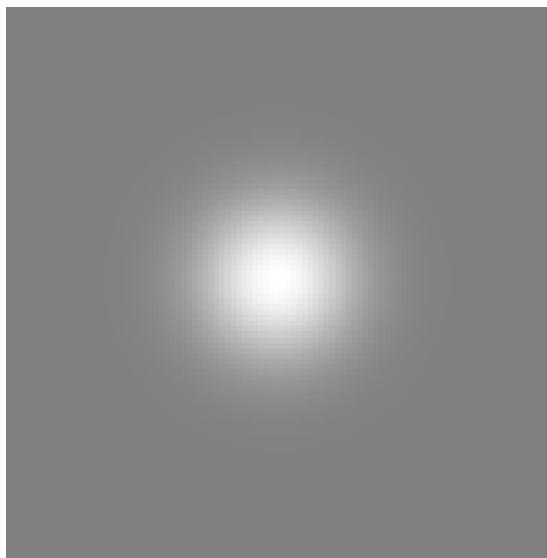
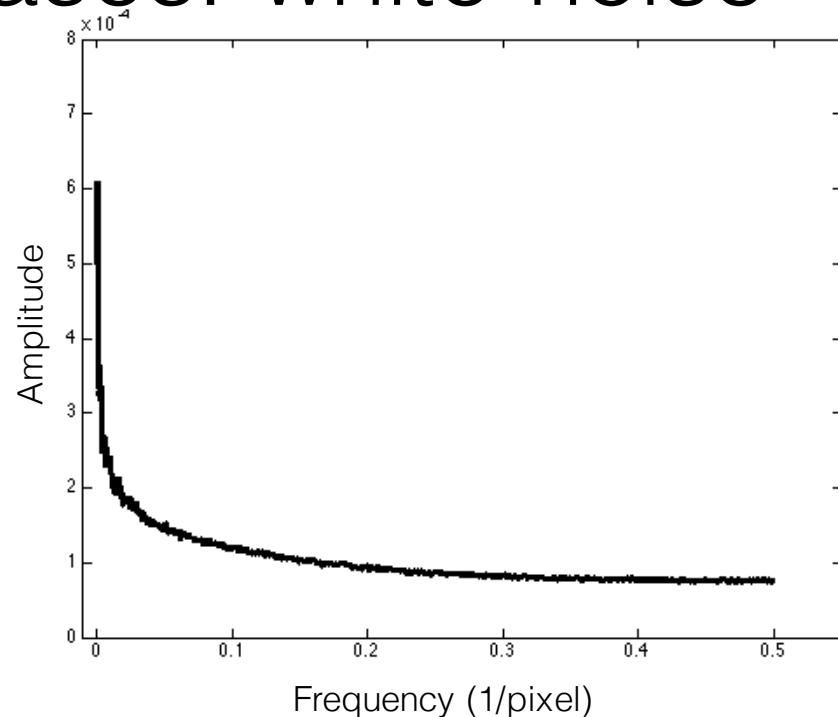
```
cutoff      = 0.15
bandwidth   = .5
[num, den]  = iirnotch(cutoff, bandwidth);
denoised_im = filter(num, den, image);
```



More difficult cases: white noise



More difficult cases: white noise



Gaussian filter

More difficult cases: white noise



More difficult cases: white noise



denoised

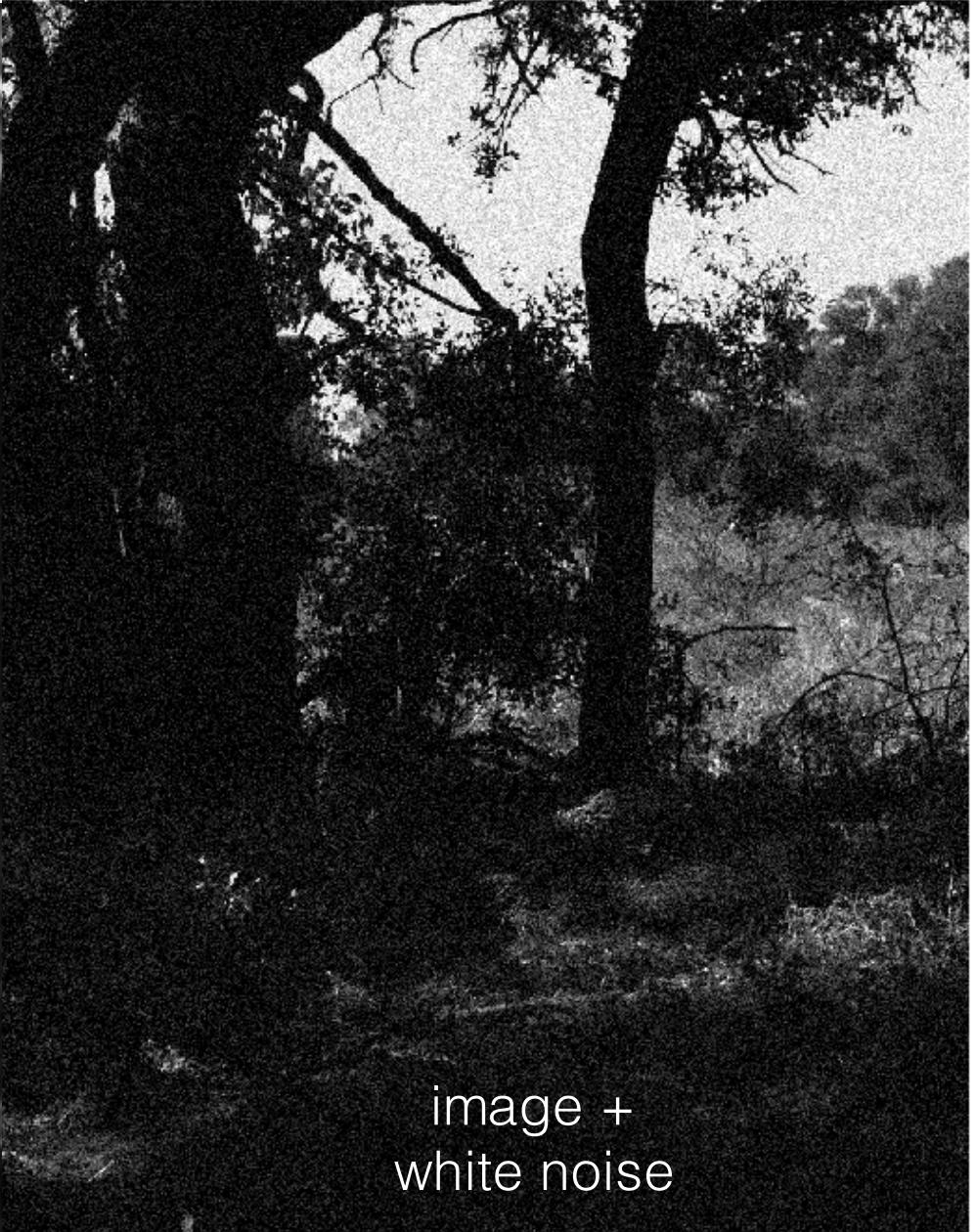
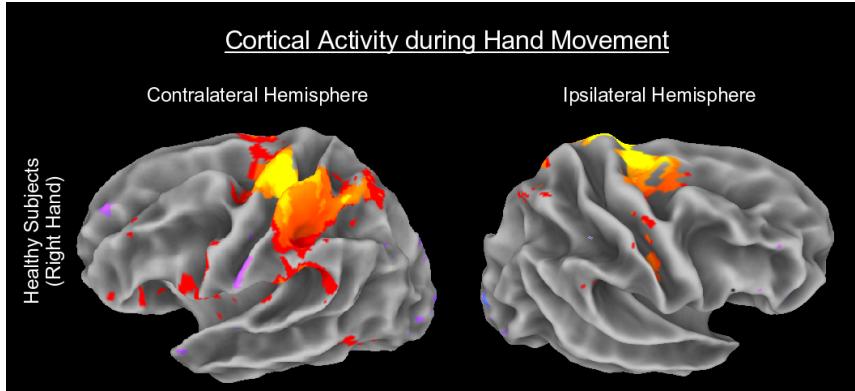
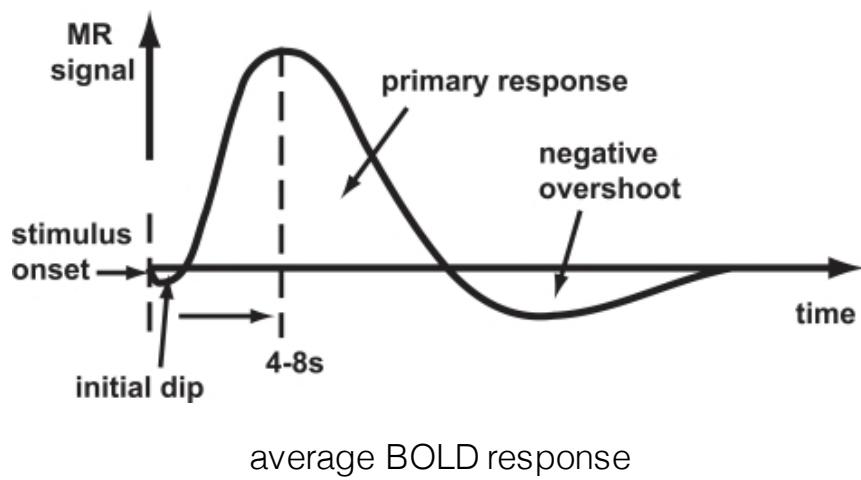
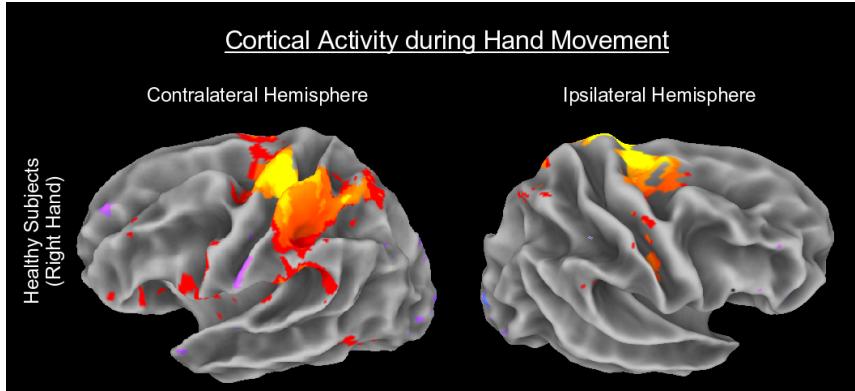


image +
white noise

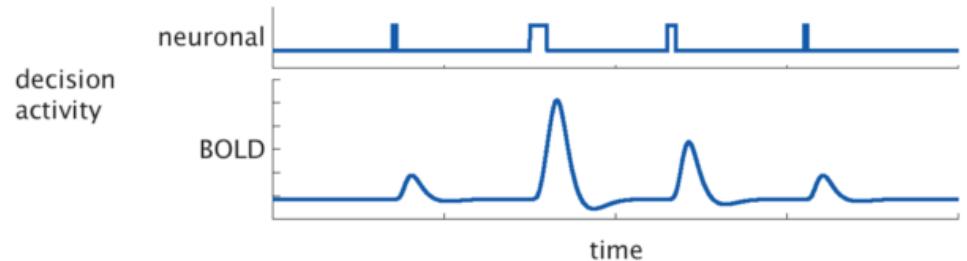
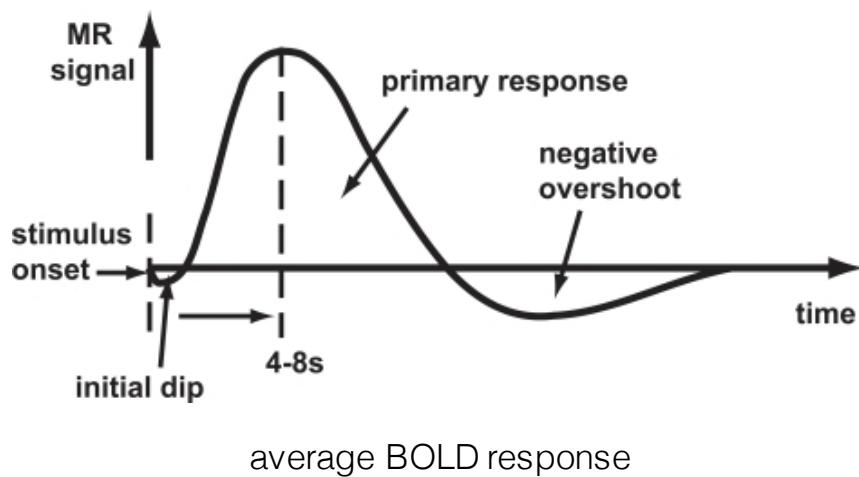
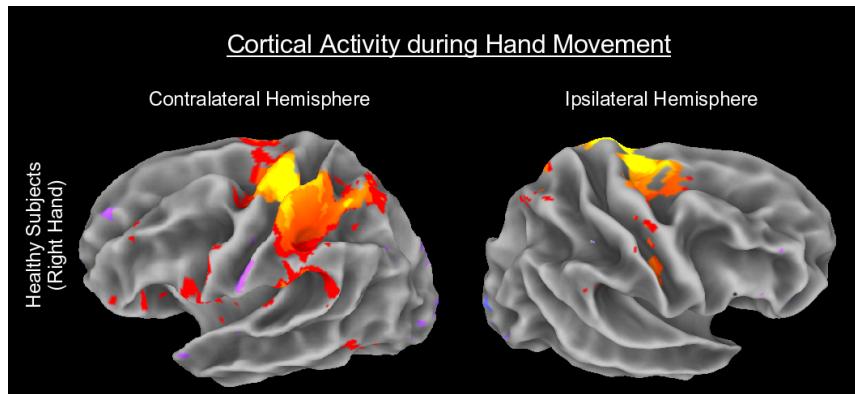
Example: Deconvolution and fMRI



Example: Deconvolution and fMRI



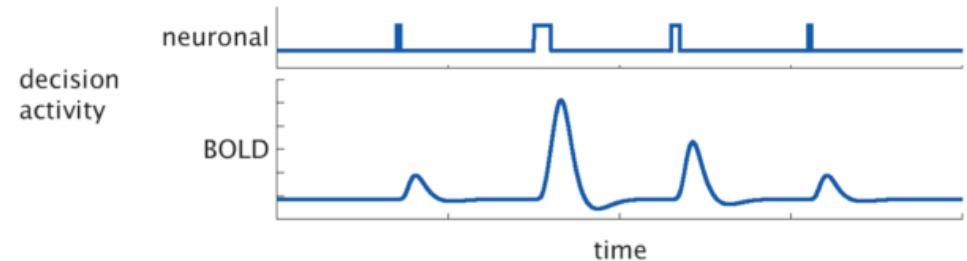
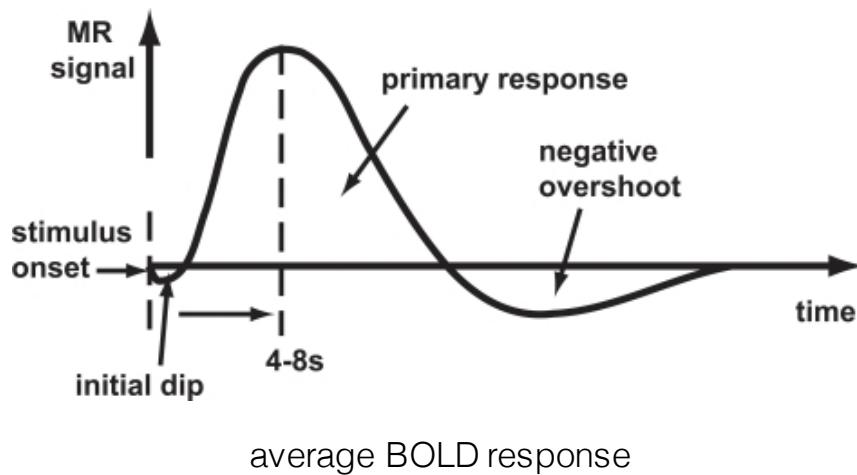
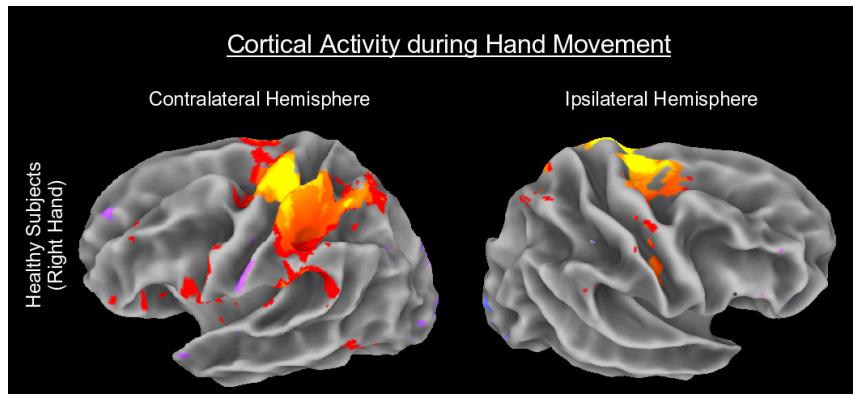
Example: Deconvolution and fMRI



BOLD signal is the convolution of the original neuronal response and the average BOLD response

How do we get back the original neuronal response?

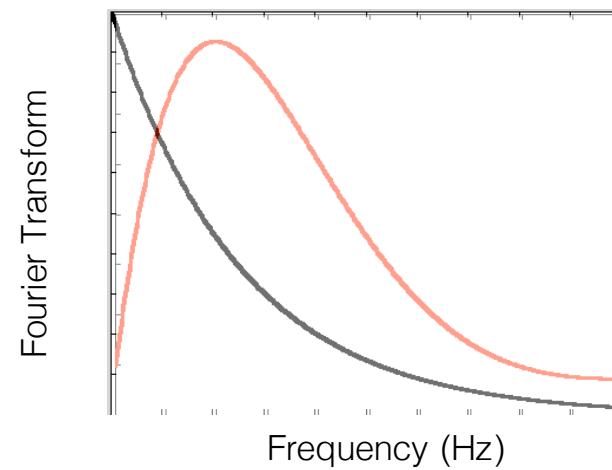
Example: Deconvolution and fMRI



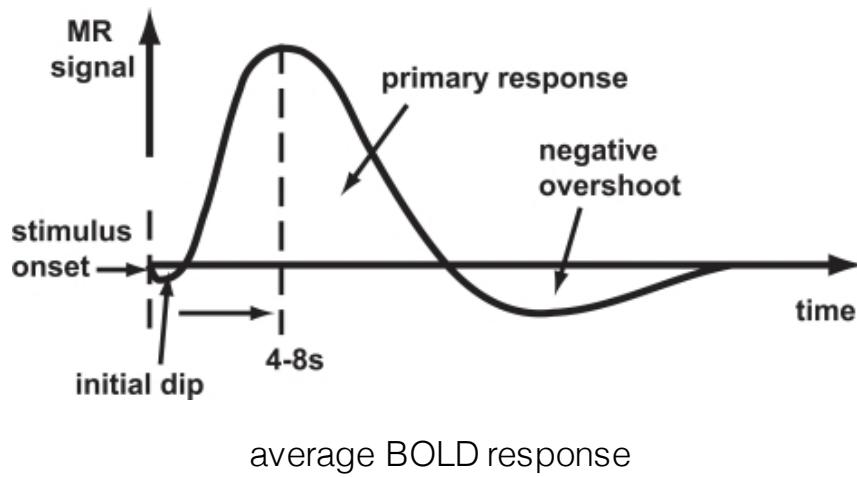
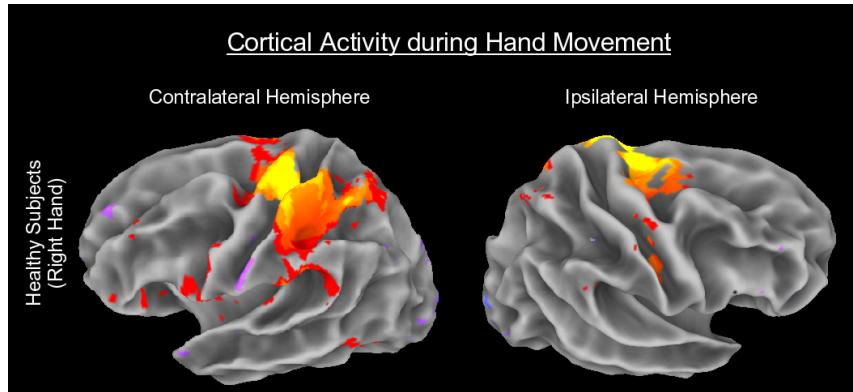
BOLD signal is the convolution of the original neuronal response and the average BOLD response

How do we get back the original neuronal response?

BOLD signal, **average BOLD response**



Example: Deconvolution and fMRI

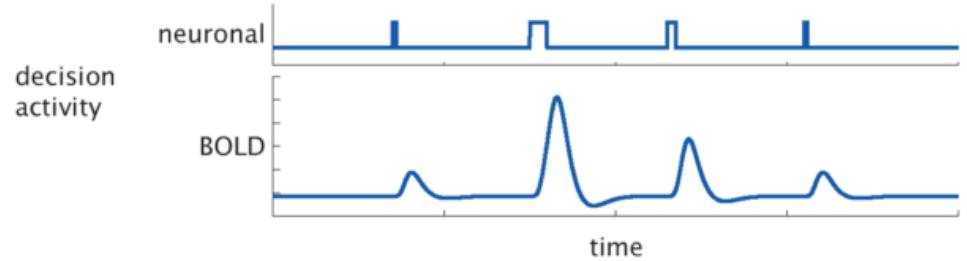


MATLAB:

```
neuronal_fft = signal_fft ./ avg_response_fft
neuronal      = ifft(neuronal_fft)
```

% alternatively, just do this:

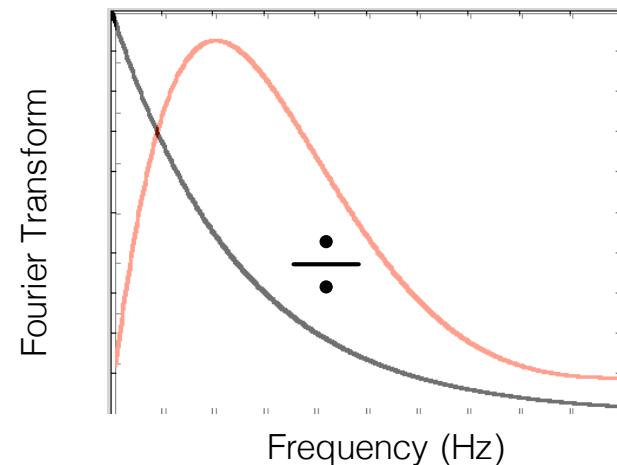
```
neuronal = deconv(signal, avg_response)
```



BOLD signal is the convolution of the original neuronal response and the average BOLD response

How do we get back the original neuronal response?

BOLD signal, **average BOLD response**



Further Reading

Analyzing behavior

Long and Fee, 2008. “Using temperature to analyse temporal dynamics in the songbird motor pathway” *Nature*

Cortical circuits

Giocomo et al. 2007. “Temporal frequency of subthreshold oscillations scales with entorhinal grid cell field spacing” *Science*

Uhlhaas and Singer, 2010. “Abnormal neural oscillations and synchrony in schizophrenia.” *Nature Neuro Reviews*

Coding

Pollen and Ronner, 1982. “Visual cortical neurons as localized spatial frequency filters.” *IEEE*

van Hateren, 1992. “Theoretical predictions of spatiotemporal receptive fields of fly LMCs, and experimental validation” *Journal of Comp. Physiol.*

Acknowledgements

Some of the slides on Fourier transforms were adapted from the original Math Tools powerpoint slides