BME I5100: Biomedical Signal Processing

Linear systems, Impulse response, Convolution, ARMA filter

Lucas C. Parra
Biomedical Engineering Department
City College of New York

CCNY

Schedule

Week 1: Introduction
Linear, stationary, normal - the stuff biology is **not** made of.

Week 1-4: Linear systems (mostly discrete time)
Impulse response
Moving Average and Auto Regressive filters
Convolution
Discrete Fourier transform and z-transform

Week 5-7: Random variables and stochastic processes
Random variables
Multivariate distributions
Statistical independence

Week 8: Electrophysiology
Origin and interpretation of Biopotentials

Week 9-14: Examples of biomedical signal processing
Probabilistic estimation
Linear discriminants - **detection** of motor activity from MEG
Harmonic analysis - **estimation** of hart rate in Speech
Auto-regressive model - **estimation** of the spectrum of 'thoughts' in EEG
Independent components analysis - **analysis** of MEG signals

Linear Time Invariant System (LTI)

A transformation L: $\mathbf{y} = L[\mathbf{x}]$ is called **linear** if:

$$\mathbf{y} = L[a\,\mathbf{x}_1 + b\,\mathbf{x}_2] = a\,L[\mathbf{x}_1] + b\,L[\mathbf{x}_2]$$

A **linear system** is a functional transformation of time functions L: $y(t)$=L[$x(t)$] such that:

$$y(t) = L[a\,x_1(t) + b\,x_2(t)] = a\,L[x_1(t)] + b\,L[x_2(t)]$$

Note that in a linear system the current output at time t may be influenced by past or future inputs $x(t')$.

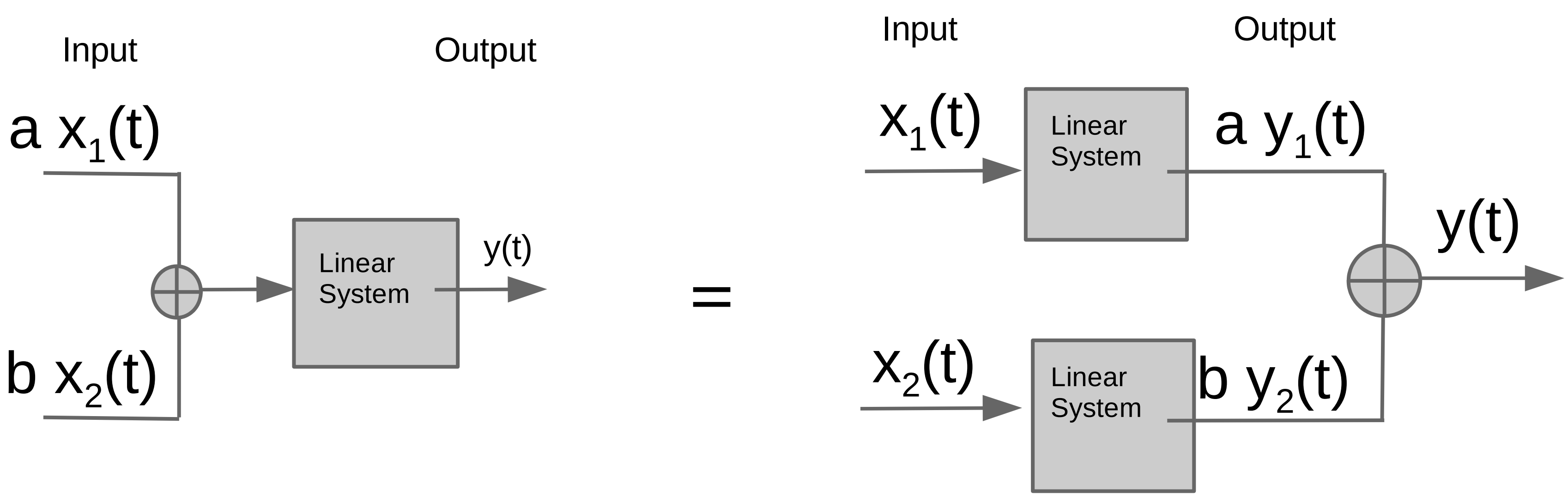
A linear system is called **time invariant** if:

$$y(t) = L[x(t)] \quad \Rightarrow \quad y(t+\tau) = L[x(t+\tau)]$$

(**shift invariant** in the discrete time case)

Linear system example

Consider a room that changes in temperature by $y(t)$ as result of heat delivered to the room as a function of time, $x(t)$. This heat could be electric heating $x_1(t)$, or gas heating $x_2(t)$ or some other form of heating. The input-output relationship is a linear system if the temperature change is the same under the following two scenarios:



Both heat sources are on simultaneously with intensity *a* and *b*, resulting in temperature change $y(t)$.

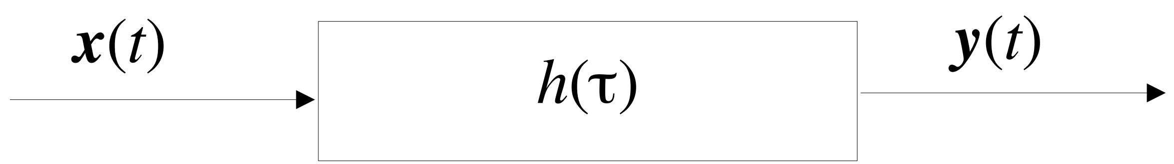
Only one heat source is on at a time, with each causing a change in temperature $y_1(t)$, or $y_2(t)$.

Impulse response

A LTI system is fully characterized by the **impulse response** $h(\tau)$. Its output is given by the convolution of the input with the impulse response (Proof from Kac Lecture.):

$$y(t) = \int_{-\infty}^{\infty} d\tau \, h(\tau) \, x(t-\tau)$$

A LTI is represented as:

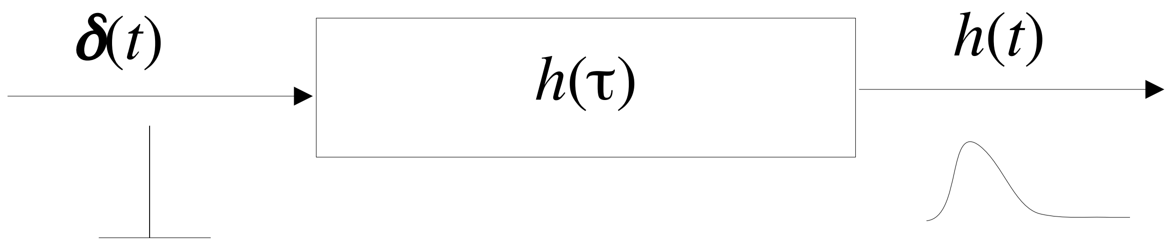


$h(\tau)$ is called impulse response because it is the system response to an input impulse:

$$x(t) = \delta(t)$$
$$y(t) = \int_{-\infty}^{\infty} d\tau \, h(\tau) \, \delta(t-\tau) = h(t)$$

Impulse response

Impulse response $h(\tau)$ can be measured using an unit impulse:

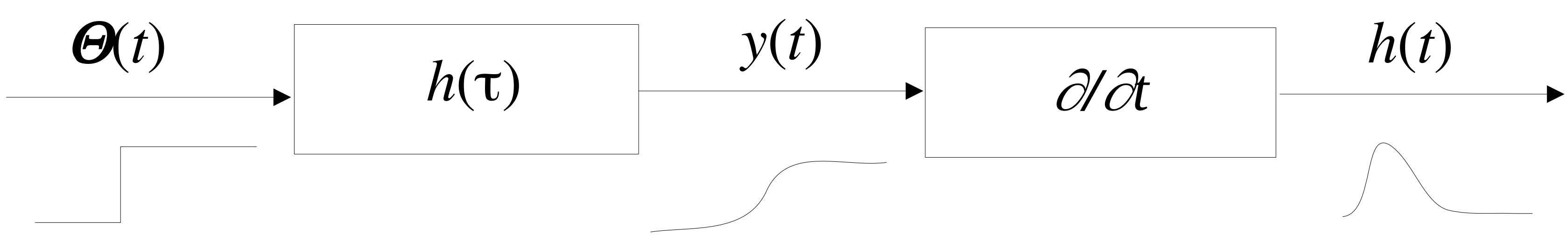


Also by differentiating the output to a step input (step response):

$$x(t) = \Theta(t) \qquad \text{where} \qquad \delta(t) = \partial/\partial t\, \Theta(t)$$

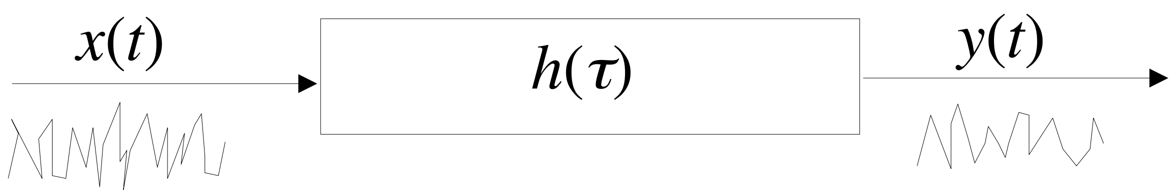
$$\frac{\partial}{\partial t} y(t) = \int_{-\infty}^{\infty} d\tau\, h(\tau) \frac{\partial}{\partial t} \Theta(t-\tau)$$

$$= \int_{-\infty}^{\infty} d\tau\, h(\tau)\, \delta(t-\tau) = h(t)$$



Impulse response -discrete, causal, finite

Often we can not control the input:



Easy to estimate $h(\tau)$ with the following simplifications:

1. **Discrete:** *Approximate* integral with sum at discrete lags $\tau = k\Delta t$
Sample input and output at times $t = n\,\Delta t$:

$$\int d\tau\, h(\tau)\, x(t-\tau) = \sum h[k]\, x[n-k]$$

2. Assume **Causal:** Depends only on the past $h[k]=0,\ k < 0$:
3. Assume **Finite Impulse Response (FIR)**: $h[k]=0,\ k > P < \infty$

$$y[n] = \sum_{k=0}^{P} h[k]\, x[n-k]$$

Impulse response estimation - MA

Leads to a **Moving Average (MA)** representation of $h[k]$:

$$y[n]=\sum_{k=0}^{Q} b[k]x[n-k]$$

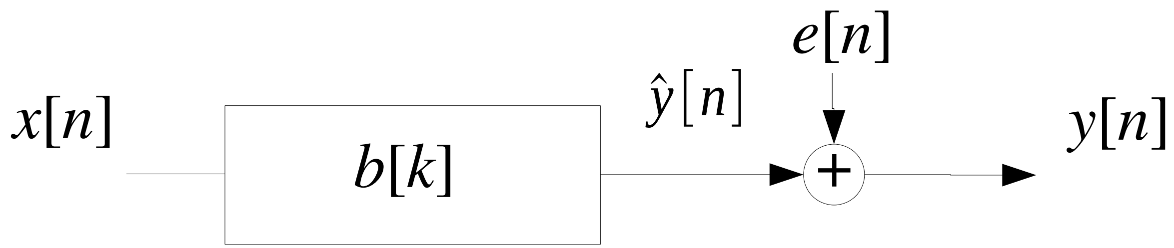
For a given Q, say Q=2, this can be rewritten as

$$\begin{bmatrix} y[1] \\ y[2] \\ y[3] \\ \dots \end{bmatrix} = \begin{bmatrix} x[1] & 0 & 0 \\ x[2] & x[1] & 0 \\ x[3] & x[2] & x[1] \\ & \dots & \end{bmatrix} \begin{bmatrix} b[0] \\ b[1] \\ b[2] \end{bmatrix} \qquad \boldsymbol{y}=\boldsymbol{X}\,\boldsymbol{b}$$

$\boldsymbol{X}$ is the **Toeplitz** matrix of $\boldsymbol{x}$ and is used to implement convolutions.

Impulse response estimation – MA model

Say $y[n]$ is observed with some added error $e[n]$. Then we can think of the MA as an estimate



$$\hat{\boldsymbol{y}} = \boldsymbol{X}\,\boldsymbol{b} \qquad \boldsymbol{y} = \hat{\boldsymbol{y}} + \boldsymbol{e}$$

For a given $\boldsymbol{y}$ and $\boldsymbol{x}$ one can identify $\boldsymbol{b}$ that minimizes the square error as the least squares solution:

$$\hat{\boldsymbol{b}} = \underset{\boldsymbol{b}}{argmin}\,\|\boldsymbol{y} - \boldsymbol{X}\,\boldsymbol{b}\|^2 = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

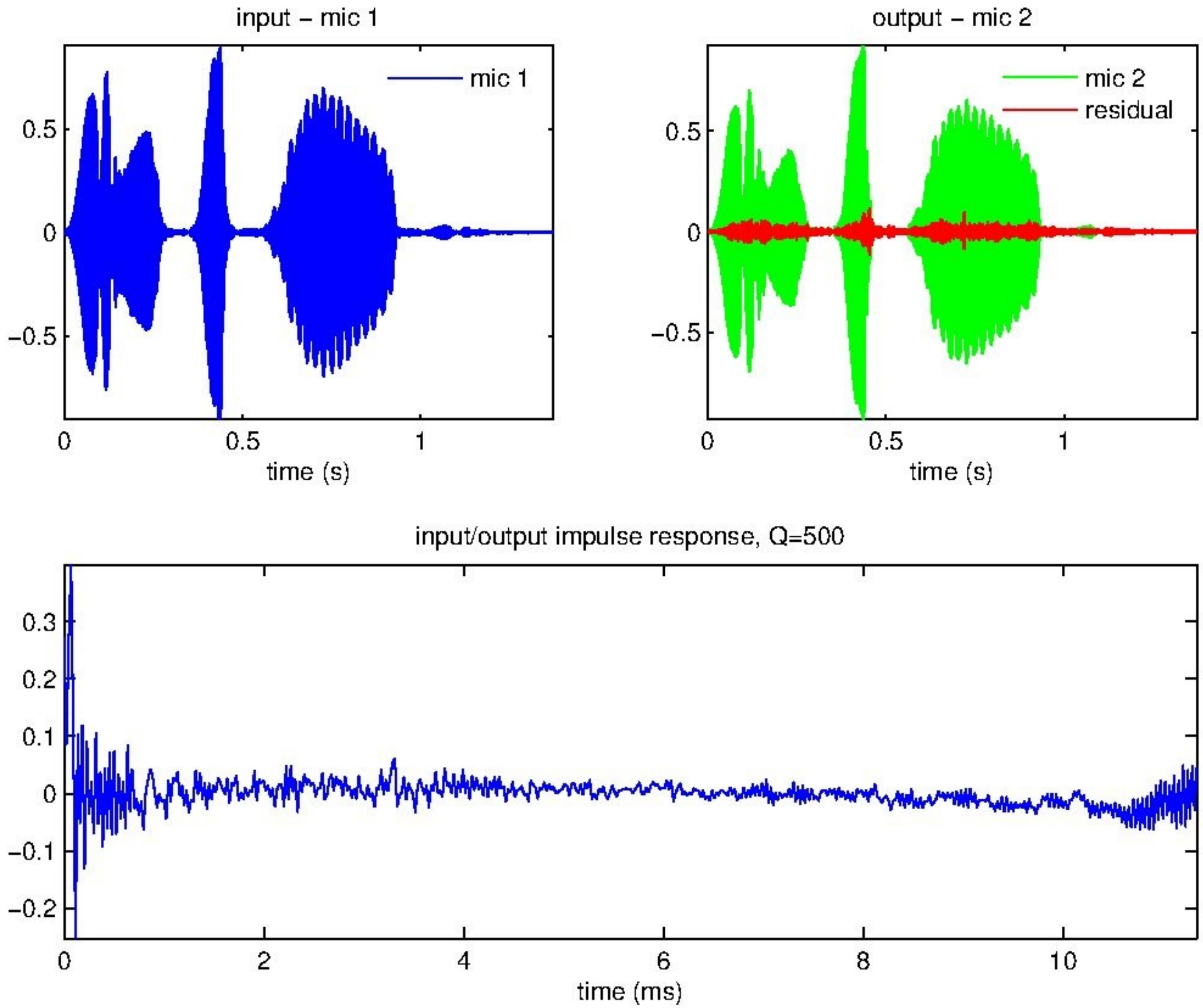
```
b = toeplitz(x,[x(1) zeros(1,Q)])\y;
```

And to avoid edge artifacts:

```
b = toeplitz(x(Q+1:end),x(Q+1:-1:1))\y(Q+1:end);
```

Impulse response - MA filter model

Example: Stereo recordings of bird song in the wild and the estimated impulse response between microphone 1 and 2.



Assignment 2: Estimate a MA filter for the relationship between one microphone and the other in `bird-stereo.wav`. Show resulting filters, input, estimated output and residual signals for varying Q. Select the "best" Q in your judgment. Estimate the microphone spacing based on the MA filter estimate.

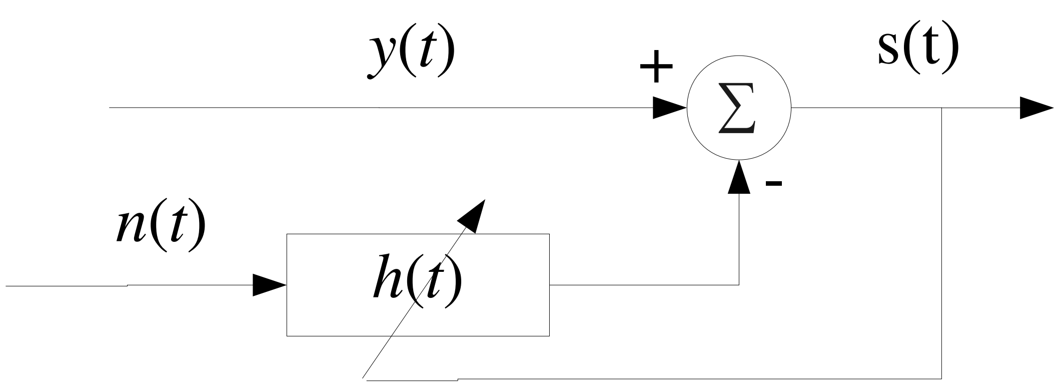
Adaptive Noise Canceling

One can use this model estimation to remove noise from data. Assume signal $x[t]$ is a noise reference signal containing only noise, and $y(t)$ is the signal of interest contaminated by noise that is distorted by some linear system

$$y(t)=s(t)+h(t)*n(t)$$

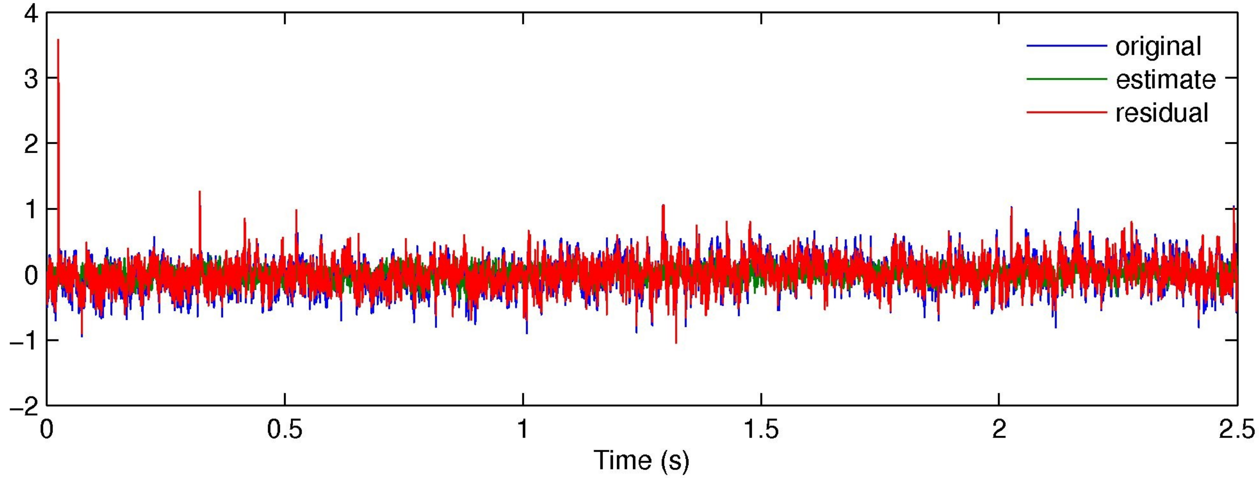
The goal of noise canceling is to find $h[t]$ so that we can recover the signal $s[t]$ by minimizing the power of $s(i)$

$$s(t)=y(t)-h(t)*n(t)$$

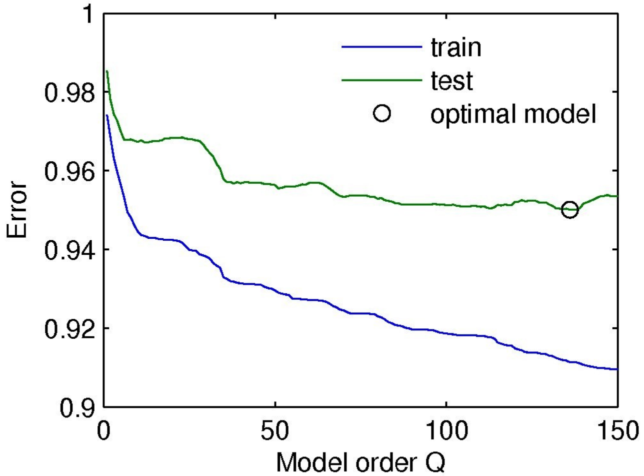


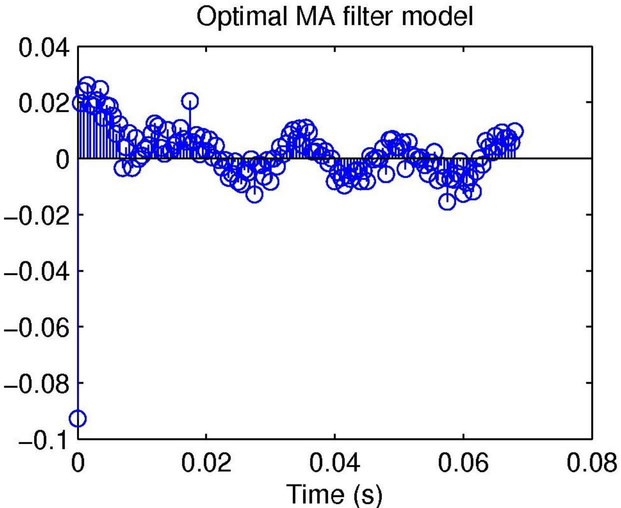
Model order selection

To select the correct 'model order' Q one typically computes model error for a training data and separate test data. For different model orders. The optimal model order is the one that minimizes the error on the test data, i.e. that best generalizes to unsee data.



In this example, local field potential data recorded in a hippocampal slice is to be predicted from a reference electrode in the slice bath but outside the tissue. This electrode only picks up noise. Therefore, the residual signal is the signal of interest.





Assignment 2 (continued): Reproduce this experiment with the data `gamma.mat`

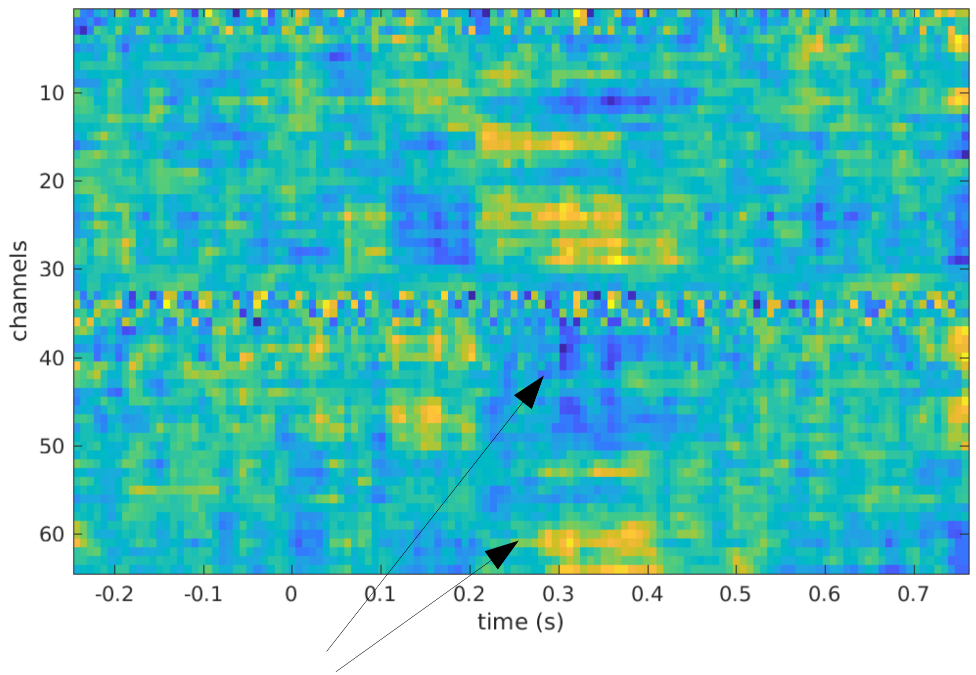
Model order selection

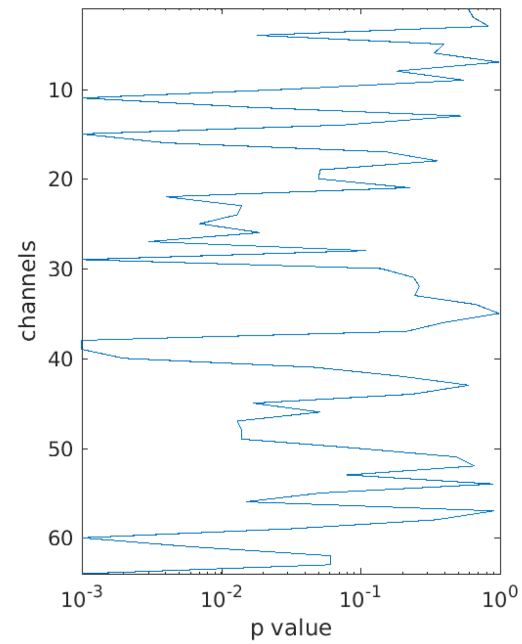
Assignment 2 (model order selection):

- Find the impulse response from one recording to the other.
- Try different lengths Q for the FIR model.
- For each Q: Compute the least squares estimate on the training data and test performance on test data, using 5 fold cross validation.
- Display train and test set performance as function of model order.
- Determine the model order with the lowest test set error and display the corresponding FIR and residual, as well as predicted an estimated output.

Temporal Response Function

The response of EEG to sound envelope can be estimated with the same approach as the impulse response, which is often referred to as Temporal Response Function.





Here it is estimated from 500s of simultaneous audio and EEG recording from one subject, displayed as an image for all 64 channels. Noticed that we estimated response prior to time zero, by shifting input relative to output by 250ms. This data is saved as audio_eeg.mat. The sound envelope can be simply estimated by taking absolute value and smoothing. If the EEG is at slower resolution then on can just down sampling to the lower EEG sampling rate:

```
>> envelope = resample(abs(audio),length(eeg),length(audio));
```

Convolution implementation

Assignment: convolution implementation

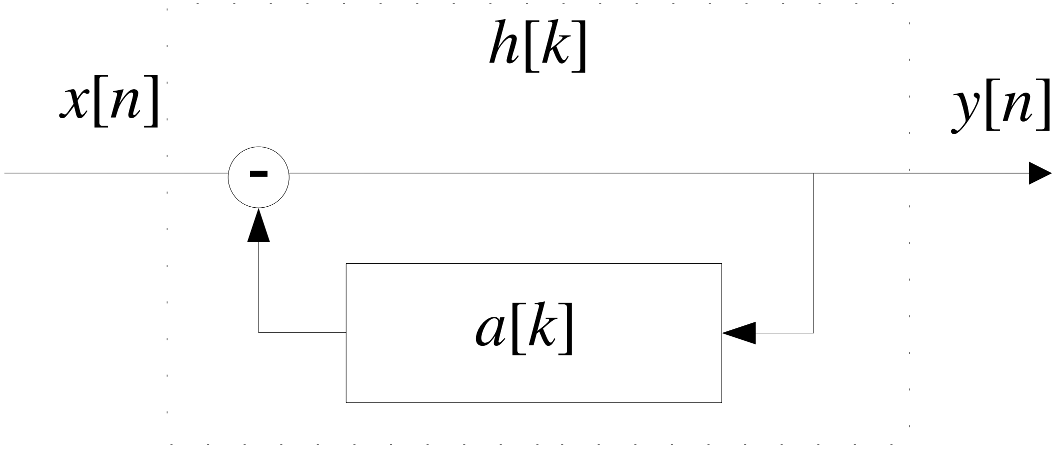
- Implement an MA with filter, conv, toepliz and explicitly with a for loop (example below) generating “valid” and “full” samples at the output, which means the output will have length (L-max(P,Q)+1) or L respectively, with L being the length of the input signal and Q,P the filter order.
- Implement the ARMA equations with filter(), and explicitly with a for loop generating “same” samples at the output. You can assume that the “history” of input and output are zero.

```
L=100; Q=3; x = randn(L,1); b = rand(Q+1,1);

% example: "same" implementation of MA
y = zeros(size(x));   % initialize sums with zero
for n=1:length(x)     % for all output samples
    for k=1:length(b) % sum over delays
        if n-k+1>0    % handle the starting edge
            y(n) = y(n) + b(k)*x(n-k+1);
        end
    end

end
```

Impulse response - discrete, causal, *infinite*

In case of **Infinite Impulse Response (IIR)** it may be beneficial to represent $h[l]$ indirectly with an **Auto Regressive (AR)** filter:

$$y[n] = x[n] - \sum_{k=1}^{P} a[k]\, y[n-k]$$



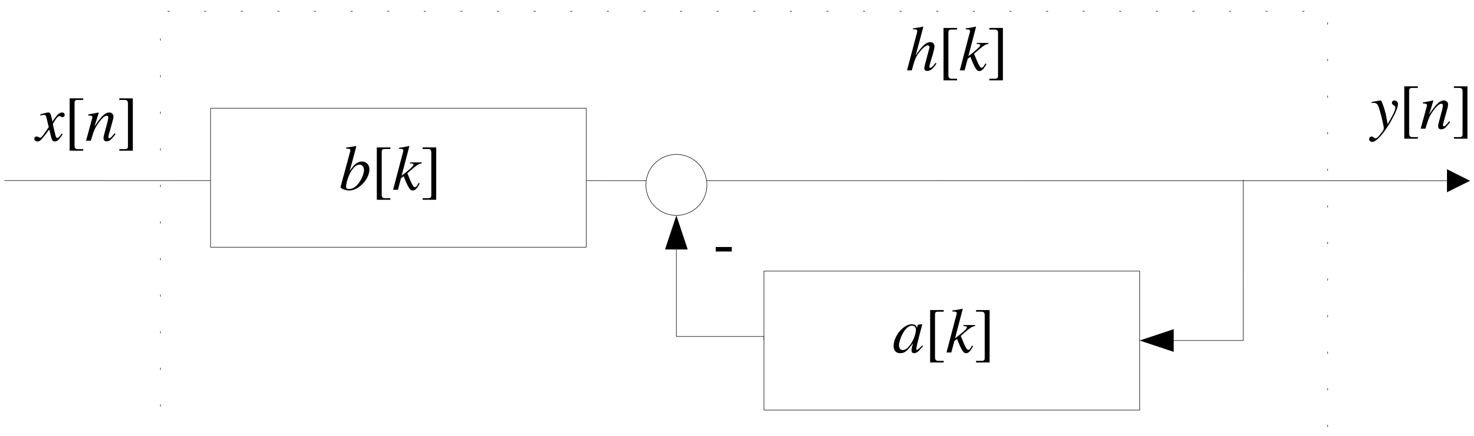
However, $h[t]$ **may not be stable!** Filter $h[k]$ is stable if:

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

Impulse response - ARMA filter

More generally an **Infinite Impulse Response (IIR)** can be represented by an **ARMA** filter (also called difference equation):

$$y[n] = -\sum_{k=1}^{P} a[k]\, y[n-k] + \sum_{k=0}^{Q} b[k]\, x[n-k]$$

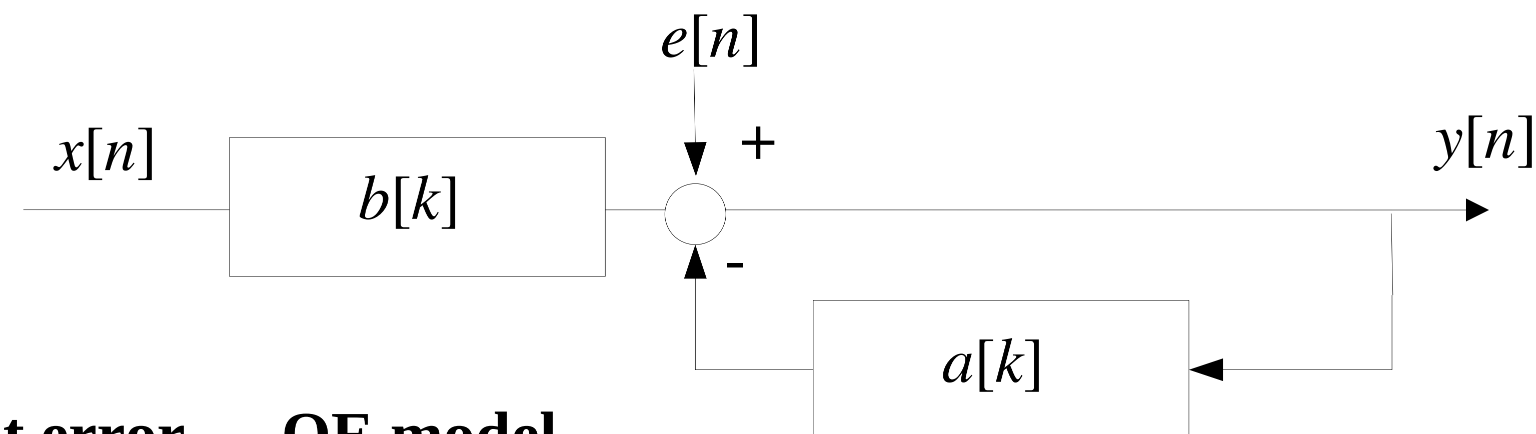


Since ARMA filter is LSI there is a corresponding *h*[*k*] that characterizes the system impulse response.

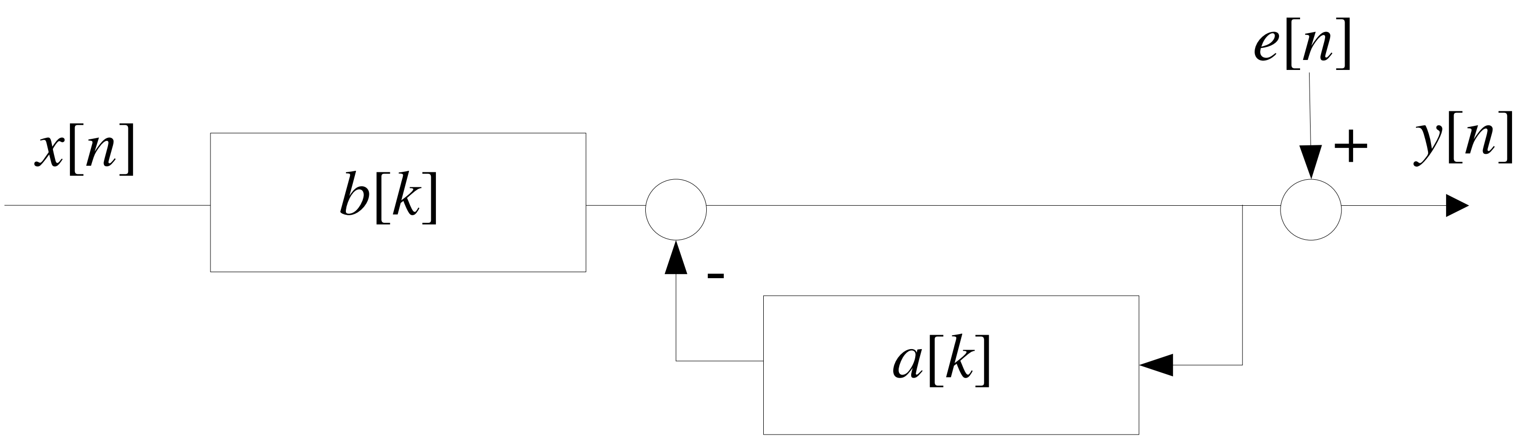
ARMA model: Equation error vs Output error

Fitting and ARMA filter to input-output data is more complicated than for an MA filter. Fitting depends what source of error is assumed:

Equation error → AR with external input (ARX) model



Output error → OE model



ARMA model: Equation error vs Output error

$y[n]$ is observable
$\hat{y}[n]$ is a model estimate

$$y[n] = \hat{y}[n] + e[n]$$

ARX model – linear in a[k], there is no recursion:

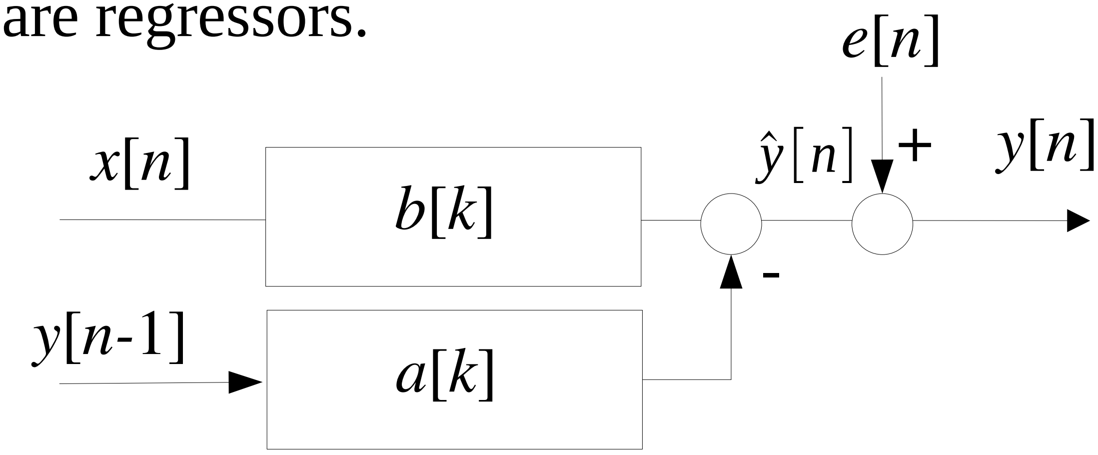
$$\hat{y}[n] = -\sum_{k=1}^{P} a[k]\, y[n-k] + \sum_{k=0}^{Q} b[k]\, x[n-k]$$

OE model – non-linear in a[k], there is a feedback recursion:

$$\hat{y}[n] = -\sum_{k=1}^{P} a[k]\, \hat{y}[n-k] + \sum_{k=0}^{Q} b[k]\, x[n-k]$$

ARX model

Error is added inside of the recursion, and preceding *y*[*n*] are observable. So it can used are part of the regression. Minimizing square error results in simple least-square linear regression with input and previous output are regressors.



$$y[n]=\hat{y}[n]+e[n]$$

$$\hat{y}[n]=-\sum_{k=1}^{P} a[k]\,y[n-k]+\sum_{k=0}^{Q} b[k]\,x[n-k]$$

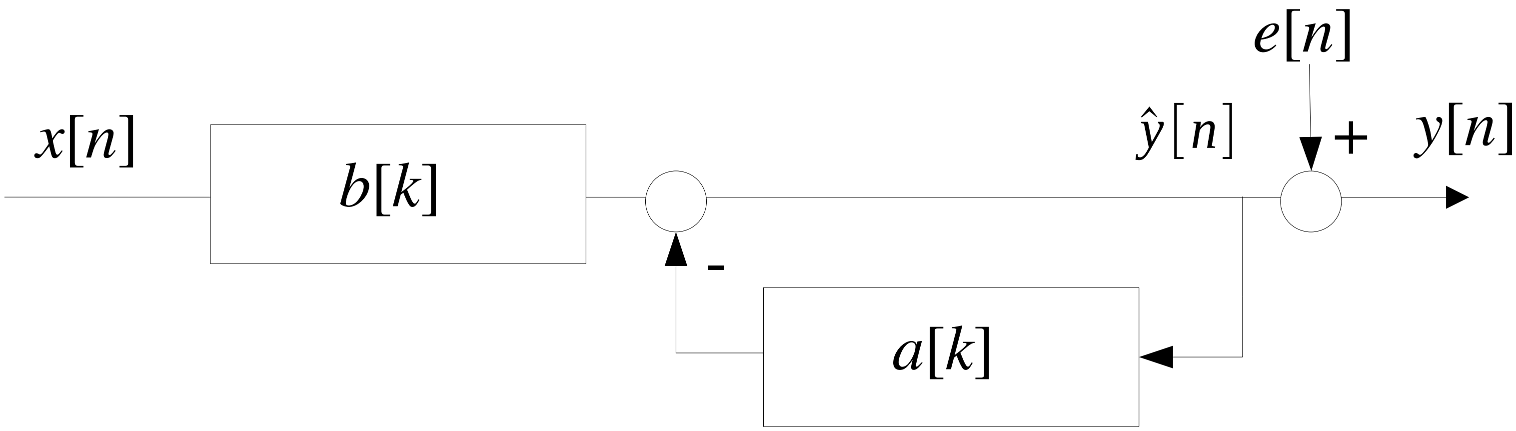
Same approach as in MA filter, but now concatenating Toeplix matrix X with a Toepliz matrix Y of *y*[t-1], *y*[*t*-2] … *y*[*t*-*P*].

```
ba = [X Y]\y;
```

… or in matlab: arx.m

Output Error model

Error is added after the ARMA structure, and ARMA filter output is not observable. Now parameters $a[k]$ contribute non-linearly to a minimum error criterion.



$$\hat{y}[n] = -\sum_{k=1}^{P} a[k]\,\hat{y}[n-k] + \sum_{k=0}^{Q} b[k]\,x[n-k]$$

Can be solved with error-back propagation (Shynk 1989) or general purpose non-linear optimization (Ljung 1999) … in matlab: oe.m

John Shynk, 1989, Adaptive IIR filter, IEEE ASSP Magazine
Lennart Ljung. System Identification: Theory for the User, 2nd edition: Prentice-Hall PTR, 1999.

Impulse response - ARMA filter

Assignment 3:
A) Show that an ARMA filter is a linear system. You may assume $y[n]=0$ for $n<1$, i.e. Zero memory as initial condition.
B) Optional question: Is an ARMA filter shift invariant?

Hint for part A: Use proof by induction. Base the induction by proving linearity for $n<1$. In the induction step assume linearity for $n-1$, $n-2$, $n-3$, ... and then prove it for n using the definition of the ARMA filter.

Definitions:

$$y[n]=L[x[n]]=-\sum_{k=1}^{P} a[k]y[n-k]+\sum_{k=0}^{Q} b[k]x[n-k]$$

$$y_1[n]=L[x_1[n]]=\ldots$$

$$y_2[n]=L[x_2[n]]=\ldots$$

Linearity: With $x[n]=c_1 x_1[n]+c_2 x_2[n]$ show that $y[n]=c_1 y_1[n]+c_2 y_2[n]$

Impulse response -ARMA filter

$$y[n] = -\sum_{k=1}^{P} a[k]\, y[n-k] + \sum_{k=0}^{Q} b[k]\, x[n-k]$$

```
>> y = filter(b,a,x);
```

Note the ambiguity in this representation of the impulse response. For example one can represent $h(k) = c^k$ as

$$b(k) = c^k,\ Q \to \infty \text{ and } a{=}0,\ P = 0$$

or as

$$b(0) = 1,\ Q = 0 \text{ and } a(1) = -c,\ P = 1$$

Advantage of the AR representation: Smaller number of parameters.

Disadvantage: Stability in not guaranteed! Test with

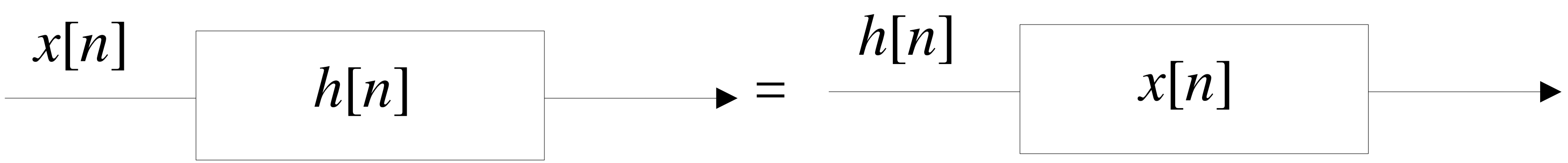
```
>> abs(roots(a))<1
```

Convolution

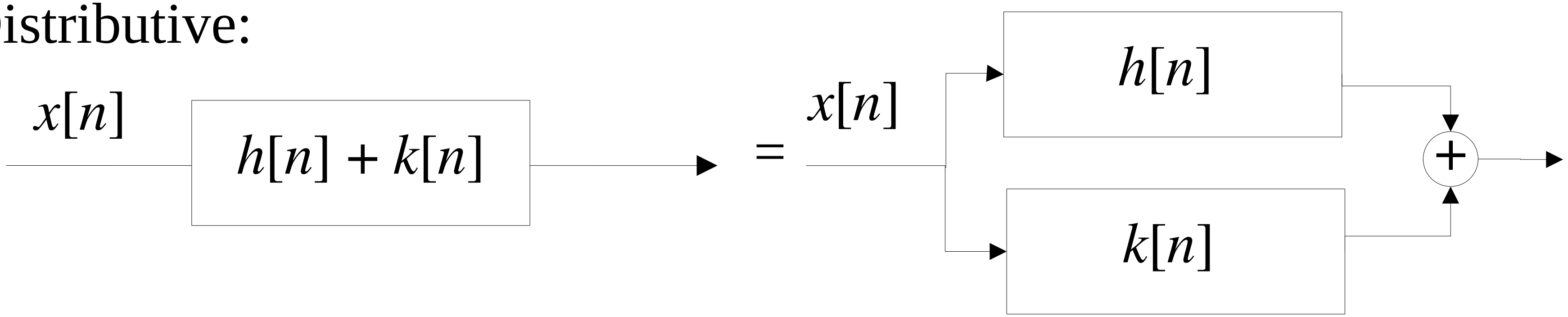
$$h[n] * x[n] = \sum_{k=-\infty}^{\infty} h[k]\,x[n-k]$$

Using this definition one can show the following properties:

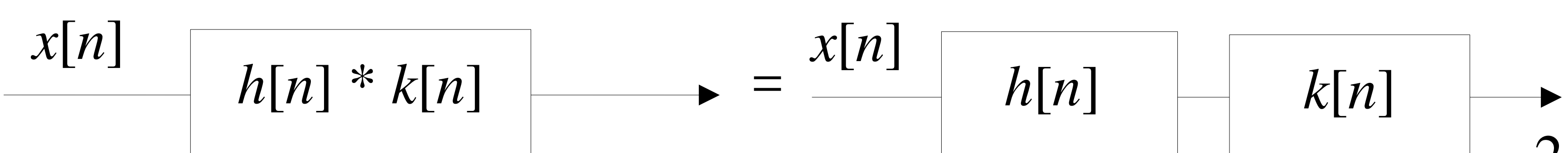
Commutative:



Distributive:



Associative:



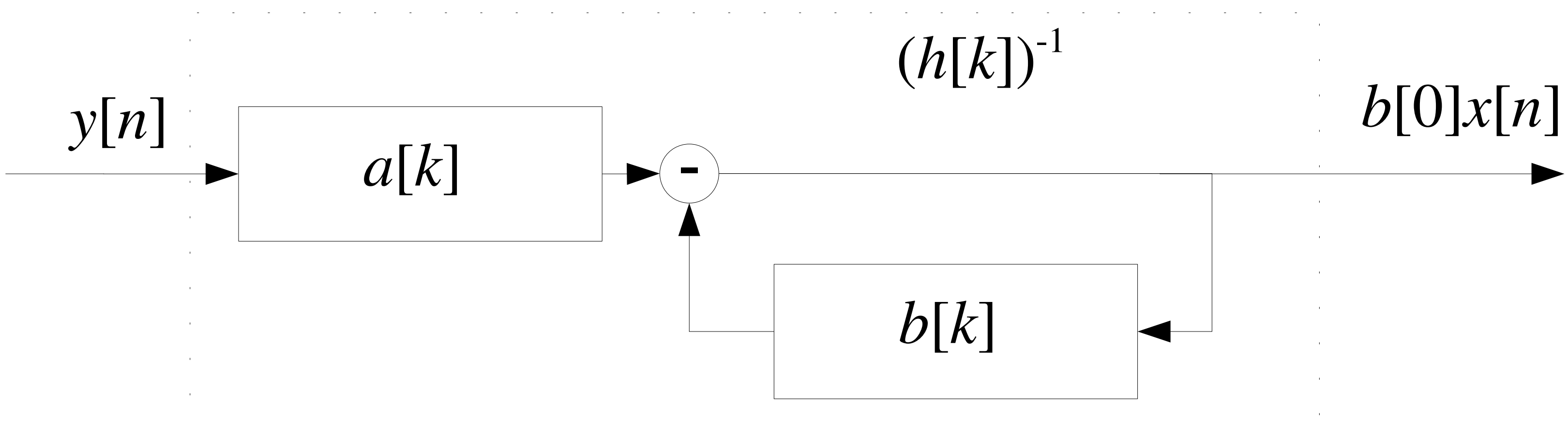
Impulse response - ARMA inverse

Notice the symmetry of the ARMA filter definition:

$$\sum_{k=0}^{P} a[k]\,y[n-k]=\sum_{k=0}^{Q} b[k]\,x[n-k]$$

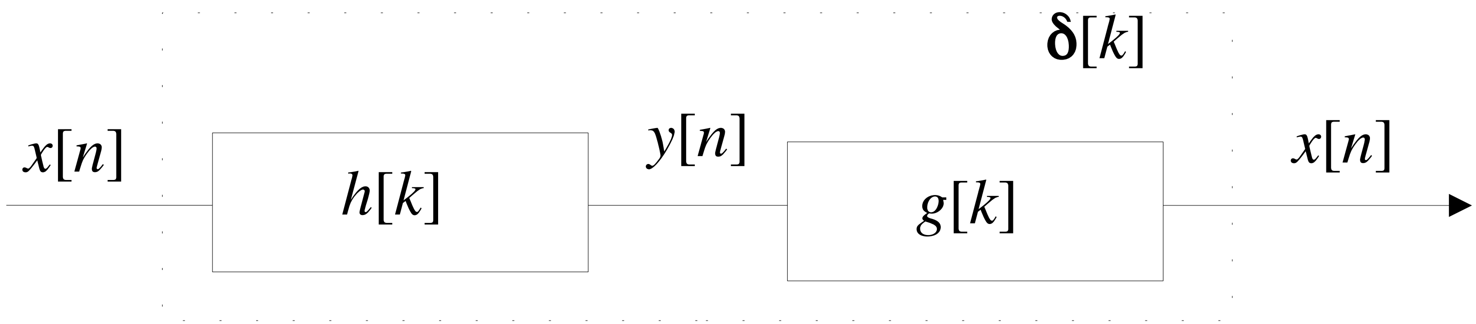
where $a[0]=1$. The inversion of $h[n]$ is given then by

$$b[0]\,x[n]=-\sum_{k=1}^{Q} b[k]\,x[n-k]+\sum_{k=0}^{P} a[k]\,y[n-k]$$



Impulse response - FIR inverse

What is the **causal inverse** $g[k]$ to a **causal FIR** $h[k]$?



Since convolutions is associative:
$$\delta[n] = \sum_{k=0}^{Q} h[k]\, g[n-k]$$

After rearranging terms:
$$g[n]h[0] = \delta[n] - \sum_{k=1}^{Q} h[k]\, g[n-k]$$

Therefore, the causal inverse of a causal FIR filter is the impulse response to the corresponding AR filter: