Adaptive Filters – Algorithms (Part 1)

Gerhard Schmidt

Christian-Albrechts-Universität zu Kiel
Faculty of Engineering
Institute of Electrical and Information Engineering
Digital Signal Processing and System Theory

Today

Contents of the Lecture:

Exercises:

- Topics for the Talks

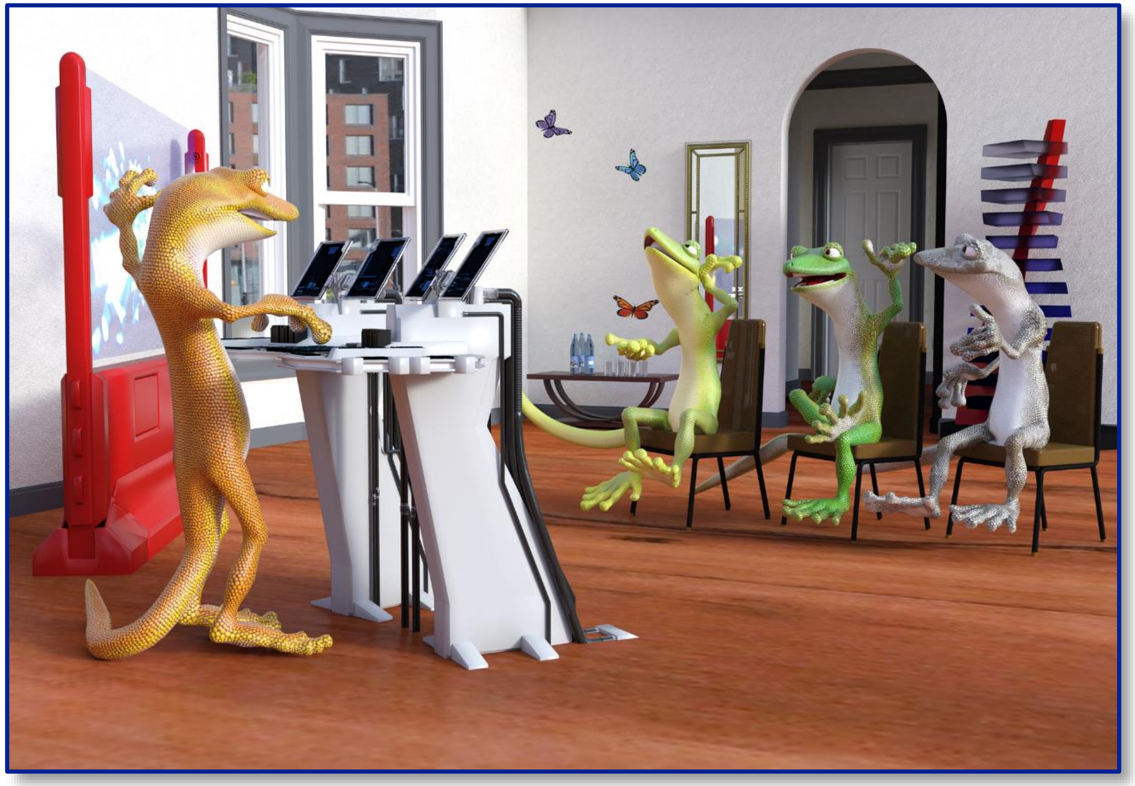
Adaptive Algorithms:

- Introductory Remarks
- Recursive Least Squares (RLS) Algorithm
- Least Mean Square Algorithm (LMS Algorithm) – Part 1
- Least Mean Square Algorithm (LMS Algorithm) – Part 2
- Affine Projection Algorithm (AP Algorithm)
- Fast Affine Projection

Possible Topics

Suggestions:

- Hearing aids
- GSM (source) coding
- Localization and tracking
- Active noise control (anti-noise)
- Noise suppression
- Adaptive beamforming
- Non-linear echo cancellation
- Feedback suppression
- …



Your own topic suggestions are welcome …

Contents

Exercises:

Adaptive Algorithms:

Motivation

Why adaptive filters?

- Signal properties are not known in advance or are time variant.
- System properties are not known in advance or time variant.

Examples:

- Speech signals
- Mobile telephone channels

Literature

Books:

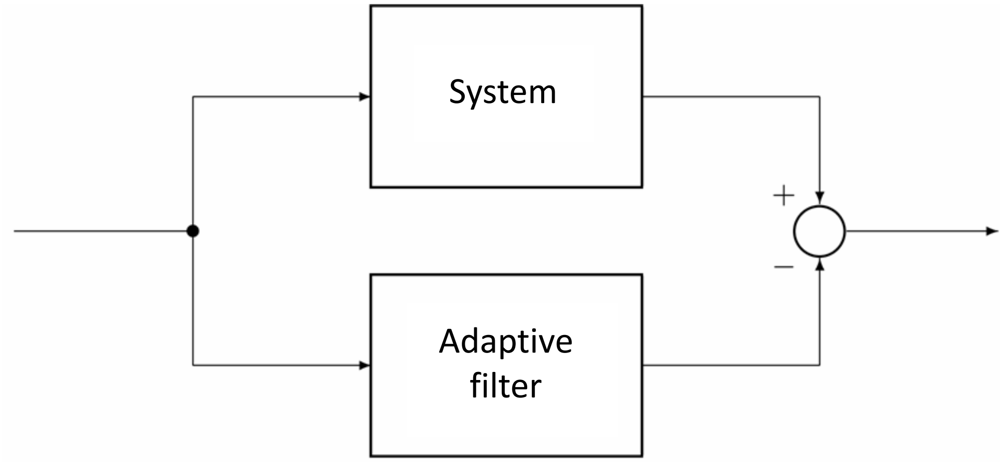
- E. Hänsler, G. Schmidt: *Acoustic Echo and Noise Control*, Wiley, 2004
- S. Haykin: *Adaptive Filter Theory*, Prentice Hall, 2002
- A. Sayed: *Fundamentals of Adaptive Filtering*, Wiley, 2004
- E. Hänsler: *Statistische Signale: Grundlagen und Anwendungen*, Springer, 2001 (in German)

Christian-Albrechts-Universität zu Kiel

Two Hook-Ups of Adaptive Filters

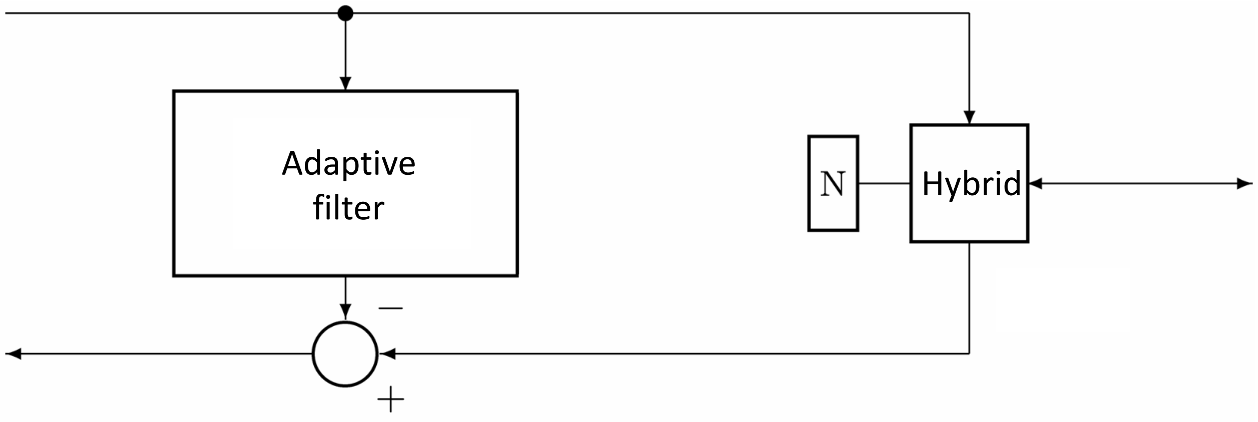
Adaptive filter for channel equalization:

Adaptive filter for system identification:



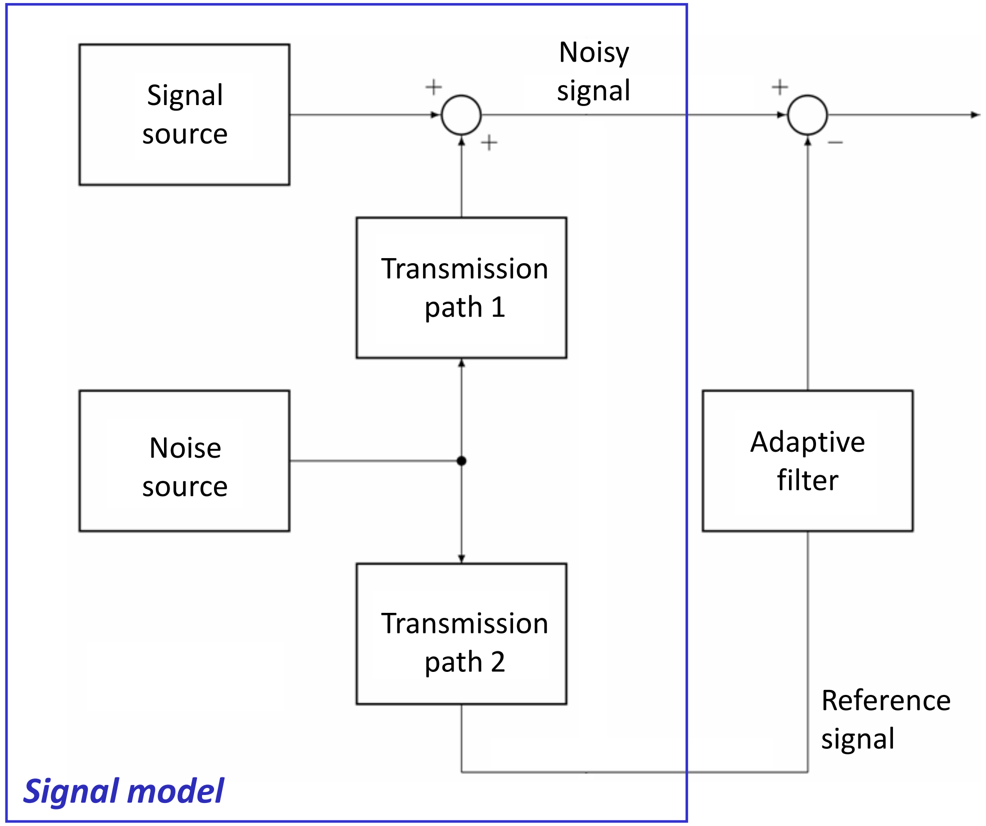
Application Examples – Part 1

Adaptive filter for cancellation of hybrid echoes:



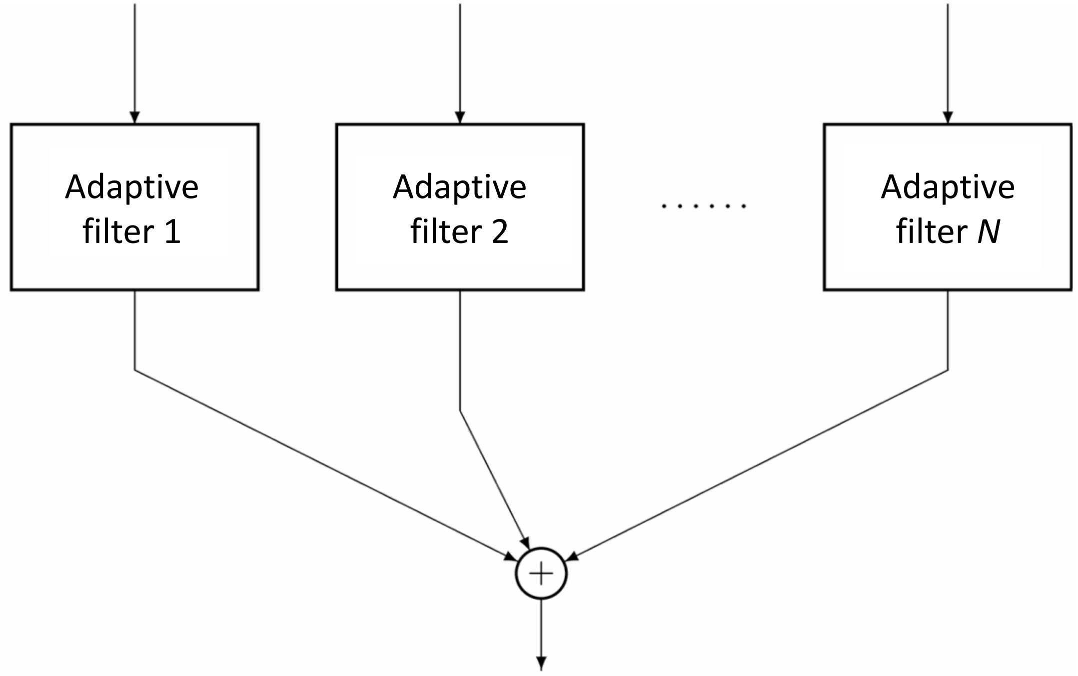
Application Examples – Part 2

Adaptive filter for noise reduction with reference signal:



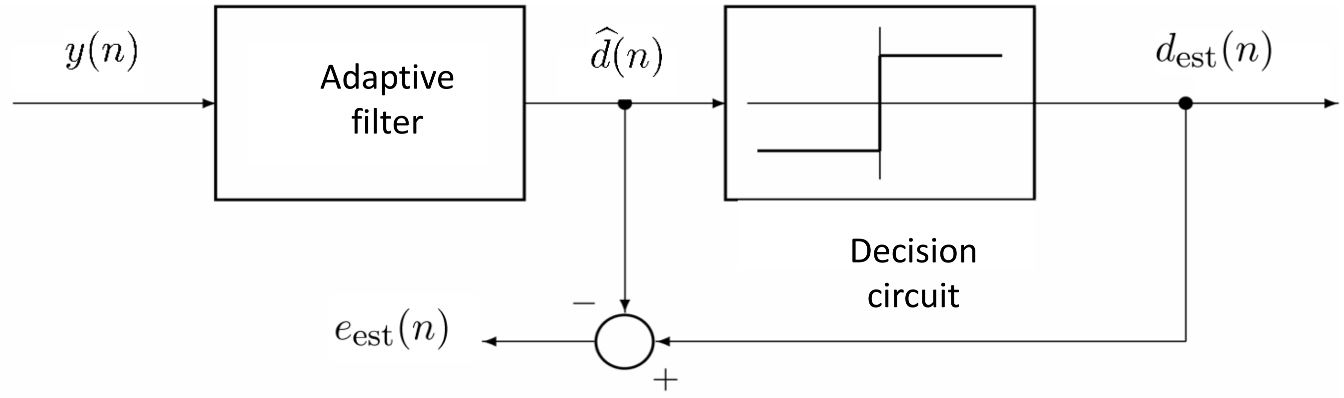
Application Examples – Part 3

Antenna array:



Application Examples – Part 4

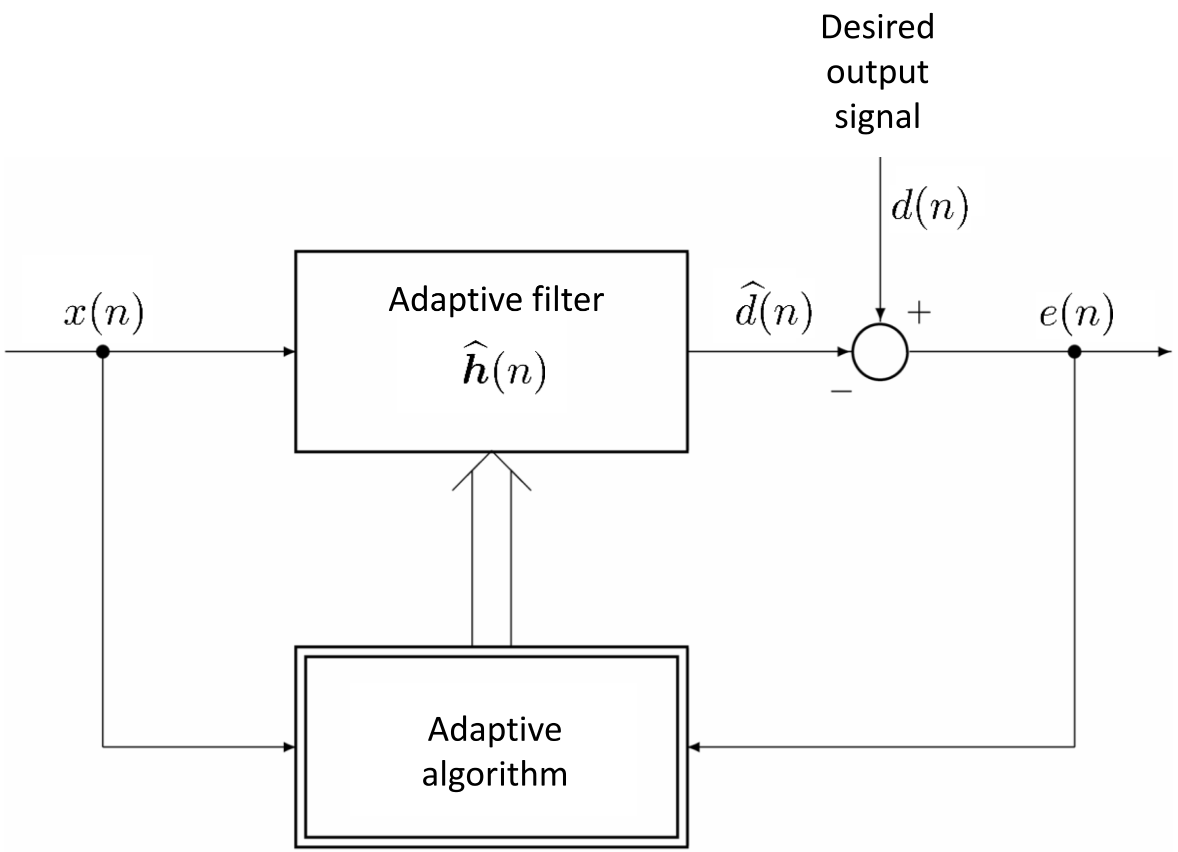
Adaptive equalization without reference signal



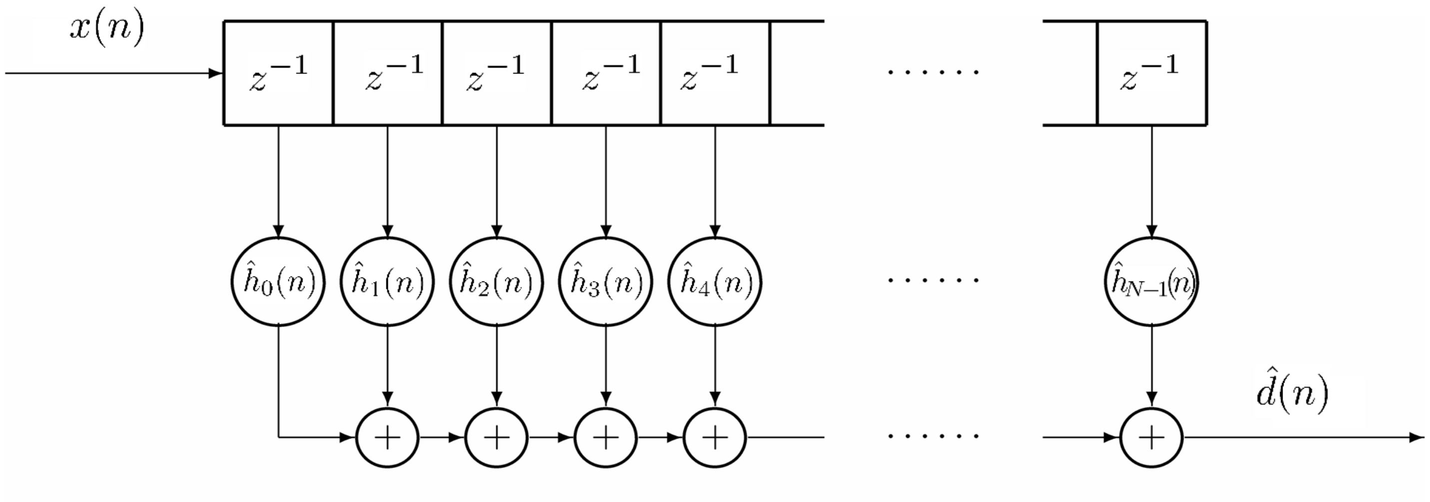
Assumptions:

$$\begin{aligned} d_{\mathrm{est}}(n) &\approx d(n) \\ e_{\mathrm{est}}(n) &\approx e(n) \end{aligned}$$

Generic Setup



Structure of an Adaptive FIR Filter

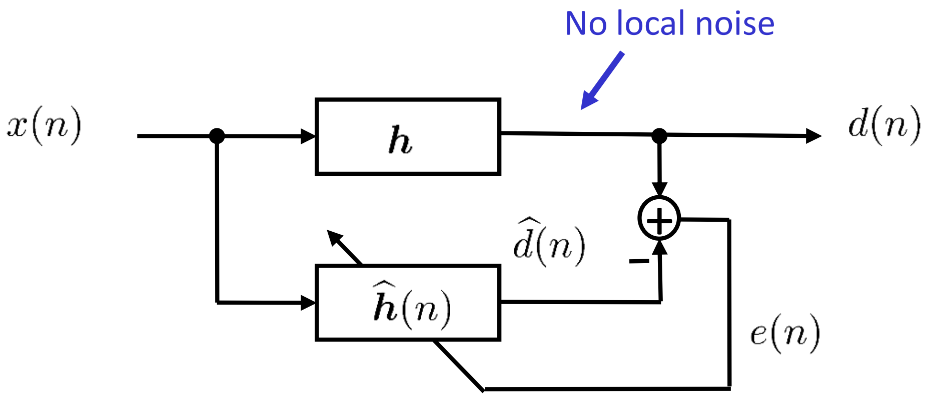


$$\boldsymbol{x}(n) = \Big[x(n),\, x(n-1),\, x(n-2),\, \ldots,\, x(n-N+1)\Big]^{\mathrm{T}}$$

$$\widehat{\boldsymbol{h}}(n) = \Big[\widehat{h}_0(n),\, \widehat{h}_1(n),\, \widehat{h}_2(n),\, \ldots,\, \widehat{h}_{N-1}(n)\Big]^{\mathrm{T}}$$

$$\widehat{d}(n) = \widehat{\boldsymbol{h}}^{\mathrm{H}}(n)\,\boldsymbol{x}(n) = \boldsymbol{x}^{\mathrm{T}}(n)\,\widehat{\boldsymbol{h}}^{*}(n)$$

Error Measures – Part 1



Mean square (signal) error:

$$\mathrm{E}\left\{\left|e(n)\right|^2\right\} = \mathrm{E}\left\{\left|d(n) - \widehat{d}(n)\right|^2\right\}$$

System distance:

$$\begin{aligned}\left\|\boldsymbol{h}_\Delta(n)\right\|^2 &= \left\|\boldsymbol{h} - \widehat{\boldsymbol{h}}(n)\right\|^2 \\ &= \left[\boldsymbol{h}(n) - \widehat{\boldsymbol{h}}(n)\right]^{\mathrm{H}} \left[\boldsymbol{h}(n) - \widehat{\boldsymbol{h}}(n)\right]\end{aligned}$$

Mean Square Error and System Distance

Relation of the normalized mean square (signal) error power and the system distance:

$$\frac{\mathrm{E}\left\{\left|e(n)\right|^2\right\}}{\mathrm{E}\left\{\left|x(n)\right|^2\right\}} = \frac{\left[\boldsymbol{h}(n) - \widehat{\boldsymbol{h}}(n)\right]^{\mathrm{H}} \mathrm{E}\left\{\boldsymbol{x}(n)\,\boldsymbol{x}^{\mathrm{H}}(n)\right\} \left[\boldsymbol{h}(n) - \widehat{\boldsymbol{h}}(n)\right]}{\mathrm{E}\left\{\left|x(n)\right|^2\right\}}$$

$$= \frac{\boldsymbol{h}_{\Delta}^{\mathrm{H}}(n)\,\mathrm{E}\left\{\boldsymbol{x}(n)\,\boldsymbol{x}^{\mathrm{H}}(n)\right\} \boldsymbol{h}_{\Delta}(n)}{\mathrm{E}\left\{\left|x(n)\right|^2\right\}}$$

Let $x(n)$ be white noise:

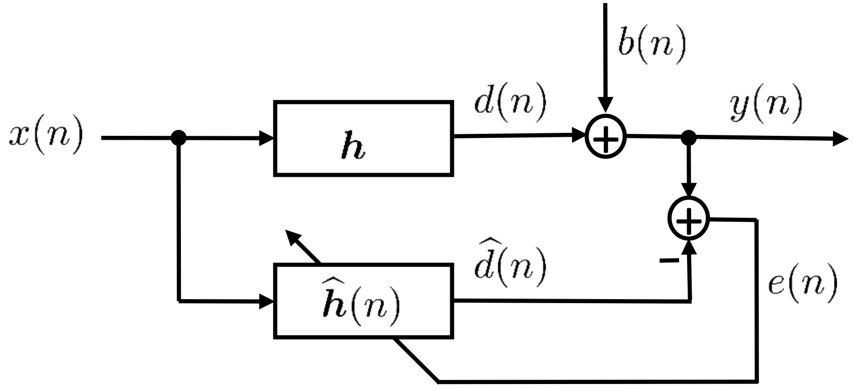
$$\mathrm{E}\left\{\boldsymbol{x}(n)\,\boldsymbol{x}^{\mathrm{H}}(n)\right\} = \text{unit matrix} \times \mathrm{E}\left\{\left|x(n)\right|^2\right\}$$

$$\frac{\mathrm{E}\left\{\left|e(n)\right|^2\right\}}{\mathrm{E}\left\{\left|x(n)\right|^2\right\}} = \boldsymbol{h}_{\Delta}^{\mathrm{H}}(n)\,\boldsymbol{h}_{\Delta}(n)$$

Adaptation

Basic principle:

New = old + correction



Properties:

- „Correction" depends on the input signal $x(n)$ and the error signal $e(n)$.
- Procedures differ by the functions $g\big(\boldsymbol{x}(n)\big)$ and $f\big(e(n)\big)$:

$$\widehat{\boldsymbol{h}}(n+1) \;=\; \widehat{\boldsymbol{h}}(n) + \mu\, \boldsymbol{x}(n)\, g\big(\boldsymbol{x}(n)\big)\, f\big(e(n)\big)\,.$$

↑

Step size

Error Measures

Three error measures control the adaptation:

- Coefficient error

$$\boldsymbol{h}_{\Delta}(n) \;=\; \boldsymbol{h}(n) - \widehat{\boldsymbol{h}}(n)$$

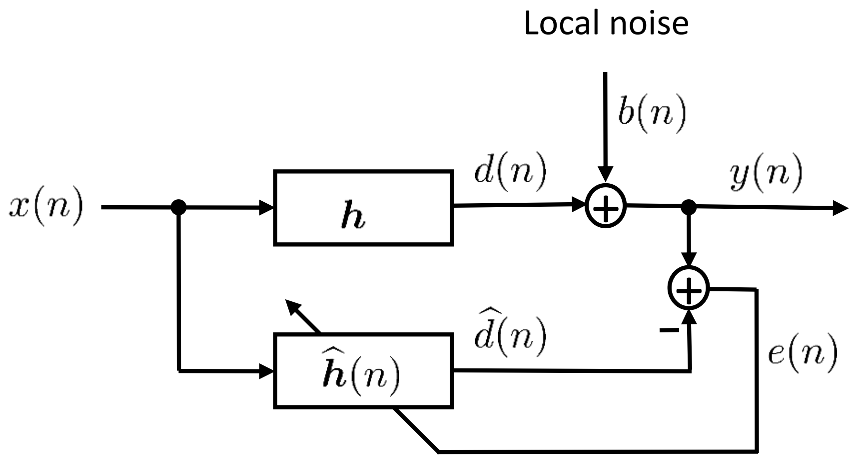
- A priori error

$$e(n|n) \;=\; \boldsymbol{h}_{\Delta}^{\mathrm{H}}(n)\,\boldsymbol{x}(n) + b(n)$$

- A posteriori error:

$$e(n|n+1) \;=\; \boldsymbol{h}_{\Delta}^{\mathrm{H}}(n+1)\,\boldsymbol{x}(n) + b(n)$$

Old data ***New filter***



Contents

Exercises:

Adaptive Algorithms:

Algorithmic Properties

Attributes of the RLS algorithm:

- No a priori knowledge of signal statistics is required.
- Optimization criterion is the (weighted) sum of squared errors.

Error Criterion

Signal *Filter*

$$\mathcal{E}(n) \;=\; \sum_{l=0}^{n} \lambda^{n-l}\, e(l|n)\, e^*(l|n) \;\; \text{with } 0 < \lambda \leq 1$$

Forgetting factor

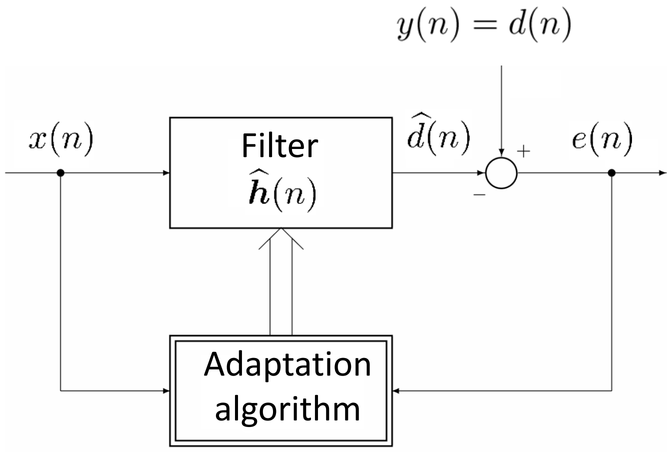
Alternative: $$\mathcal{E}(n) \;=\; \sum_{l=n-L+1}^{n} e(l|n)\, e^*(l|n)$$

Filter at time n

$$e(l|n) \;=\; y(l) - \widehat{\boldsymbol{h}}^{\mathrm{H}}(n)\, \boldsymbol{x}(l) \;=\; y(l) - \boldsymbol{x}^{\mathrm{T}}(l)\, \widehat{\boldsymbol{h}}^{*}(n)$$

Signal at time l

$$\mathcal{E}(n) \;=\; \sum_{l=0}^{n} \lambda^{n-l} \left[y(l) - \widehat{\boldsymbol{h}}^{\mathrm{H}}(n)\, \boldsymbol{x}(l)\right] \left[y^*(l) - \boldsymbol{x}^{\mathrm{H}}(l)\, \widehat{\boldsymbol{h}}(n)\right]$$



Derivation – Part 1

Cost function:

$$\mathcal{E}(n) = \sum_{l=0}^{n} \lambda^{n-l} \left[y(l) - \widehat{\boldsymbol{h}}^{\mathrm{H}}(n)\, \boldsymbol{x}(l)\right] \left[y^*(l) - \boldsymbol{x}^{\mathrm{H}}(l)\, \widehat{\boldsymbol{h}}(n)\right]$$

Differentiate with respect to the complex filter coefficients and setting the result to zero:

$$\sum_{l=0}^{n} \lambda^{n-l}\, \boldsymbol{x}(l)\, \boldsymbol{x}^{\mathrm{H}}(l)\, \widehat{\boldsymbol{h}}(n) = \sum_{l=0}^{n} \lambda^{n-l}\, \boldsymbol{x}(l)\, y^*(l)$$

Definitions:

$$\widehat{\boldsymbol{R}}_{xx}(n) = \sum_{l=0}^{n} \lambda^{n-l}\, \boldsymbol{x}(l)\, \boldsymbol{x}^{\mathrm{H}}(l)$$

… Estimate for the auto correlation matrix

$$\widehat{\boldsymbol{r}}_{xy}(n) = \sum_{l=0}^{n} \lambda^{n-l}\, \boldsymbol{x}(l)\, y^*(l)$$

… Estimate for the cross correlation vector

Derivation – Part 2

From Simon Haykin, „Adaptive Filter Theory“, Prentice Hall, 2002:

Section B.2 Examples 797

EXAMPLE 3

Consider the real-valued cost function (see Chapter 2)

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{w}^H\mathbf{p} - \mathbf{p}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}\mathbf{w}.$$

Using the results of Examples 1 and 2, we find that the conjugate derivative of J with respect to the tap-weight vector $\mathbf{w}$ is

$$\frac{\partial J}{\partial \mathbf{w}^*} = -\mathbf{p} + \mathbf{R}\mathbf{w}. \tag{B.11}$$

Let $\mathbf{w}_o$ be the optimum value of the tap-weight vector $\mathbf{w}$ for which the cost function J is minimal, or, equivalently, the derivative $(\partial J/\partial \mathbf{w}^*) = \mathbf{0}$. Then, from Eq. (B.11), we infer that

$$\mathbf{R}\mathbf{w}_o = \mathbf{p}. \tag{B.12}$$

This is the matrix form of the Wiener–Hopf equations for a transversal filter operating in a stationary environment, characterized by the correlation matrix $\mathbf{R}$ and cross-correlation vector $\mathbf{p}$.

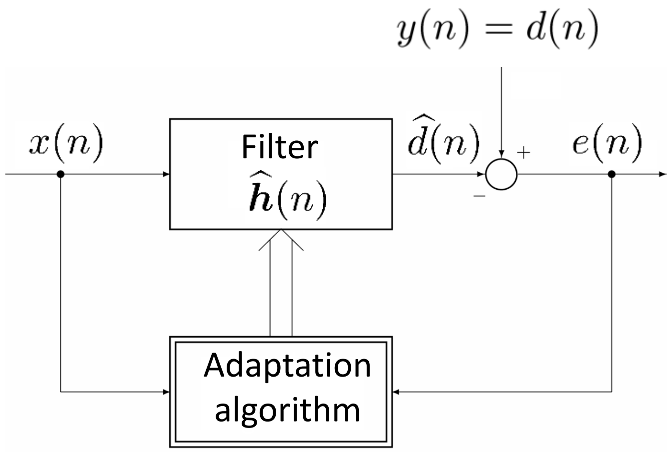
or: The Matrix Cookbook [http://matrixcookbook.com]

Derivation – Part 3

$$\sum_{l=0}^{n} \lambda^{n-l}\,\boldsymbol{x}(l)\,\boldsymbol{x}^{\mathrm{H}}(l)\,\widehat{\boldsymbol{h}}(n) = \sum_{l=0}^{n} \lambda^{n-l}\,\boldsymbol{x}(l)\,y^*(l)$$

$$\widehat{\boldsymbol{R}}_{xx}(n) = \sum_{l=0}^{n} \lambda^{n-l}\,\boldsymbol{x}(l)\,\boldsymbol{x}^{\mathrm{H}}(l)$$

$$\widehat{\boldsymbol{r}}_{xy}(n) = \sum_{l=0}^{n} \lambda^{n-l}\,\boldsymbol{x}(l)\,y^*(l)$$



Inserting the results leads to:

$$\widehat{\boldsymbol{R}}_{xx}(n)\,\widehat{\boldsymbol{h}}(n) = \widehat{\boldsymbol{r}}_{xy}(n)$$

„Wiener solution“

… assuming that the auto correlation matrix is invertible

$$\widehat{\boldsymbol{h}}(n) = \widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\widehat{\boldsymbol{r}}_{xy}(n)$$

Recursion – Part 1

Recursion of the auto correlation matrix over time:

$$\begin{aligned}\widehat{\boldsymbol{R}}_{xx}(n+1) &= \sum_{l=0}^{n+1} \lambda^{n+1-l}\, \boldsymbol{x}(l)\, \boldsymbol{x}^{\mathrm{H}}(l)\\ &= \lambda \sum_{l=0}^{n} \lambda^{n-l}\, \boldsymbol{x}(l)\, \boldsymbol{x}^{\mathrm{H}}(l) + \boldsymbol{x}(n+1)\, \boldsymbol{x}^{\mathrm{H}}(n+1)\\ &= \lambda\, \widehat{\boldsymbol{R}}_{xx}(n) + \boldsymbol{x}(n+1)\, \boldsymbol{x}^{\mathrm{H}}(n+1)\end{aligned}$$

Recursion of the cross correlation vector over time:

$$\begin{aligned}\widehat{\boldsymbol{r}}_{xy}(n+1) &= \sum_{l=0}^{n+1} \lambda^{n+1-l}\, \boldsymbol{x}(l)\, y^{*}(l)\\ &= \lambda \sum_{l=0}^{n} \lambda^{n-l}\, \boldsymbol{x}(l)\, y^{*}(l) + \boldsymbol{x}(n+1)\, y^{*}(n+1)\\ &= \lambda\, \widehat{\boldsymbol{r}}_{xy}(n) + \boldsymbol{x}(n+1)\, y^{*}(n+1)\end{aligned}$$

Recursion – Part 2

Recursion for the auto correlation matrix:

$$\widehat{\boldsymbol{R}}_{xx}(n+1) = \lambda\,\widehat{\boldsymbol{R}}_{xx}(n) + \boldsymbol{x}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)$$

Matrix Inversion Lemma:

$$\left[\boldsymbol{A} + \boldsymbol{u}\,\boldsymbol{v}^{\mathrm{H}}\right]^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\,\boldsymbol{u}\,\boldsymbol{v}^{\mathrm{H}}\,\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^{\mathrm{H}}\,\boldsymbol{A}^{-1}\,\boldsymbol{u}}$$

Inserting the Lemma in the recursion:

$$\begin{aligned}\widehat{\boldsymbol{R}}_{xx}^{-1}(n+1) &= \lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n) \\ &- \frac{\lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\lambda^{-1}}{1 + \lambda^{-1}\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)}\end{aligned}$$

Recursion – Part 3

Recursion for the auto correlation matrix:

$$\begin{aligned}\widehat{\boldsymbol{R}}_{xx}^{-1}(n+1) &= \lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n) \\ &\quad - \frac{\lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\lambda^{-1}}{1+\lambda^{-1}\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)}\end{aligned}$$

Definition of a gain vector:

$$\boldsymbol{\gamma}(n+1) = \frac{\lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)}{1+\lambda^{-1}\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)}$$

Inserting this definition leads to:

$$\widehat{\boldsymbol{R}}_{xx}^{-1}(n+1) = \lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n) - \boldsymbol{\gamma}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\lambda^{-1}$$

Recursion – Part 4

Definition of a gain factor:

$$\boldsymbol{\gamma}(n+1) \;=\; \frac{\lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)}{1+\lambda^{-1}\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)}$$

Multiplication by the denominator on the right hand side leads to:

$$\boldsymbol{\gamma}(n+1)\Big[1+\lambda^{-1}\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)\Big] \;=\; \lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)$$

Rewriting leads to:

$$\begin{aligned}\boldsymbol{\gamma}(n+1) \;&=\; \lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)\\ &\quad -\lambda^{-1}\,\boldsymbol{\gamma}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)\end{aligned}$$

$$\boldsymbol{\gamma}(n+1) \;=\; \underbrace{\Big[\lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)-\lambda^{-1}\,\boldsymbol{\gamma}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\Big]}_{\widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)}\;\boldsymbol{x}(n+1)$$

$$\boxed{\boldsymbol{\gamma}(n+1) \;=\; \widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\boldsymbol{x}(n+1)}$$

Recursion – Part 5

Recursion of the filter coefficient vector:

$$\widehat{\boldsymbol{h}}(n) = \widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\widehat{\boldsymbol{r}}_{xy}(n)$$

Step from n to n+1:

$$\widehat{\boldsymbol{h}}(n+1) = \widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\widehat{\boldsymbol{r}}_{xy}(n+1)$$

Reducing the right hand side:

$$\widehat{\boldsymbol{r}}_{xy}(n+1) = \lambda\,\widehat{\boldsymbol{r}}_{xy}(n) + \boldsymbol{x}(n+1)\,y^*(n+1)$$

Inserting the recursion of the cross correlation vector leads to:

$$\widehat{\boldsymbol{h}}(n+1) = \lambda\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\widehat{\boldsymbol{r}}_{xy}(n) + \widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\boldsymbol{x}(n+1)\,y^*(n+1)$$

Recursion – Part 6

What we have so far:

$$\widehat{\boldsymbol{h}}(n+1) \;=\; \lambda\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\widehat{\boldsymbol{r}}_{xy}(n) + \widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\boldsymbol{x}(n+1)\,y^*(n+1)\,.$$

If we insert the recursive computation of the inverse auto correlation matrix

$$\widehat{\boldsymbol{R}}_{xx}^{-1}(n+1) \;=\; \lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n) - \boldsymbol{\gamma}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\lambda^{-1}\,,$$

we obtain:

$$\begin{aligned}\widehat{\boldsymbol{h}}(n+1) \;&=\; \boxed{\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\widehat{\boldsymbol{r}}_{xy}(n)} - \boldsymbol{\gamma}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\boxed{\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\widehat{\boldsymbol{r}}_{xy}(n)}\\ &\qquad + \widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\boldsymbol{x}(n+1)\,y^*(n+1)\\ &=\; \widehat{\boldsymbol{h}}(n) - \boldsymbol{\gamma}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{h}}(n)\\ &\qquad + \widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\boldsymbol{x}(n+1)\,y^*(n+1)\,.\end{aligned}$$

Recursion – Part 7

What we have so far:

$$\begin{aligned}\widehat{\boldsymbol{h}}(n+1) &= \widehat{\boldsymbol{h}}(n) - \boldsymbol{\gamma}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{h}}(n) \\ &\quad + \widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\boldsymbol{x}(n+1)\,y^*(n+1)\,.\end{aligned}$$

Inserting $\gamma(n+1)$ ***according to***

$$\boldsymbol{\gamma}(n+1) = \widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\boldsymbol{x}(n+1)\,,$$

results in

$$\widehat{\boldsymbol{h}}(n+1) = \widehat{\boldsymbol{h}}(n) + \underbrace{\boldsymbol{\gamma}(n+1)}_{\text{Gain factor}}\underbrace{\Big[y^*(n+1) - \boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{h}}(n)\Big]}_{\text{Error: old filter with new data}}\,.$$

$$\begin{aligned}e(n+1|n) &= y(n+1) - \widehat{d}(n+1|n) \\ &= y(n+1) - \boldsymbol{x}^{\mathrm{T}}(n+1)\,\widehat{\boldsymbol{h}}^*(n)\end{aligned}$$

Adaptation Rule – Part 1

Inserting previous results:

$$\boldsymbol{\gamma}(n+1) = \widehat{\boldsymbol{R}}_{xx}^{-1}(n+1)\,\boldsymbol{x}(n+1)\,.$$

$$\widehat{\boldsymbol{h}}(n+1) = \widehat{\boldsymbol{h}}(n) + \overbrace{\boldsymbol{\gamma}(n+1)}\underbrace{\Big[y^*(n+1) - \boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{h}}(n)\Big]}\,.$$

$$y^*(n+1) - \boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{h}}(n) = e^*(n+1|n)$$

Adaptation rule for the filter coefficients according to the RLS algorithm:

$$\boxed{\widehat{\boldsymbol{h}}(n) = \widehat{\boldsymbol{h}}(n-1) + \underbrace{\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n)\,e^*(n|n-1)}_{\boldsymbol{\Delta}\widehat{\boldsymbol{h}}(n-1)}}$$

Summary

Computing a preliminary gain vector (complexity prop. N²):

$$\boldsymbol{\gamma}(n+1) \;=\; \frac{\lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)}{1+\lambda^{-1}\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\boldsymbol{x}(n+1)}$$

Update of the inverse auto correlation matrix (complexity prop. N²):

$$\widehat{\boldsymbol{R}}_{xx}^{-1}(n+1) \;=\; \lambda^{-1}\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n) - \boldsymbol{\gamma}(n+1)\,\boldsymbol{x}^{\mathrm{H}}(n+1)\,\widehat{\boldsymbol{R}}_{xx}^{-1}(n)\,\lambda^{-1}$$

Computing the error signal (complexity prop. N):

$$e(n+1|n) \;=\; y(n+1) - \widehat{\boldsymbol{h}}^{\mathrm{H}}(n)\,\boldsymbol{x}(n+1)$$

Update of the filter vector (complexity prop. N):

$$\widehat{\boldsymbol{h}}(n+1) \;=\; \widehat{\boldsymbol{h}}(n) + \mu\,\boldsymbol{\gamma}(n+1)\,e^{*}(n+1|n)$$

Step size (0 … 1), will be treated later …

Contents

Exercises:

- Topics for the Talks

Adaptive Algorithms:

- Introductory Remarks
- Recursive Least Squares (RLS) Algorithm
- ***Least Mean Square Algorithm (LMS Algorithm) – Part 1***
- Least Mean Square Algorithm (LMS Algorithm) – Part 2
- Affine Projection Algorithm (AP Algorithm)

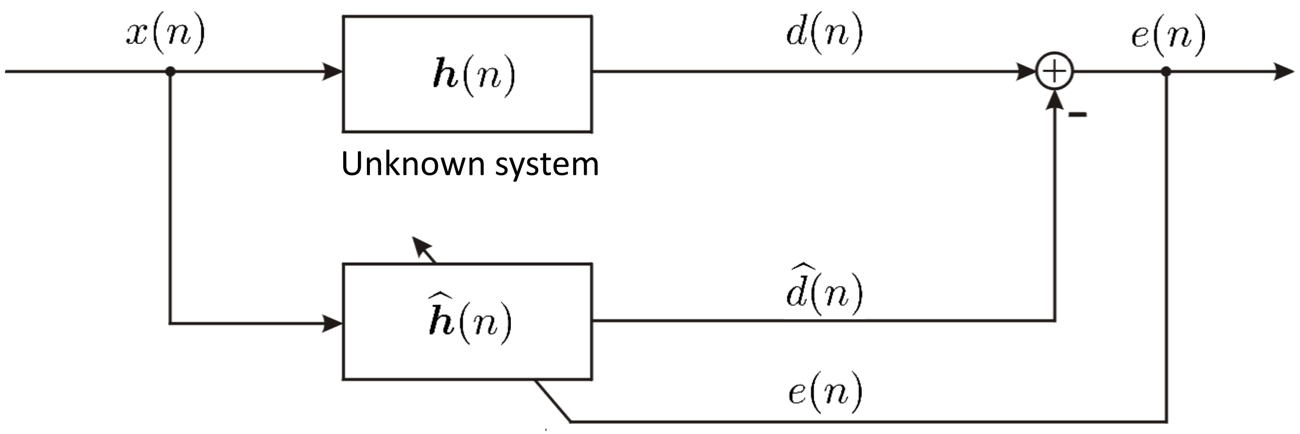
Optimization criterion:

- Minimizing the mean square error

Assumptions:

- Real, stationary random processes

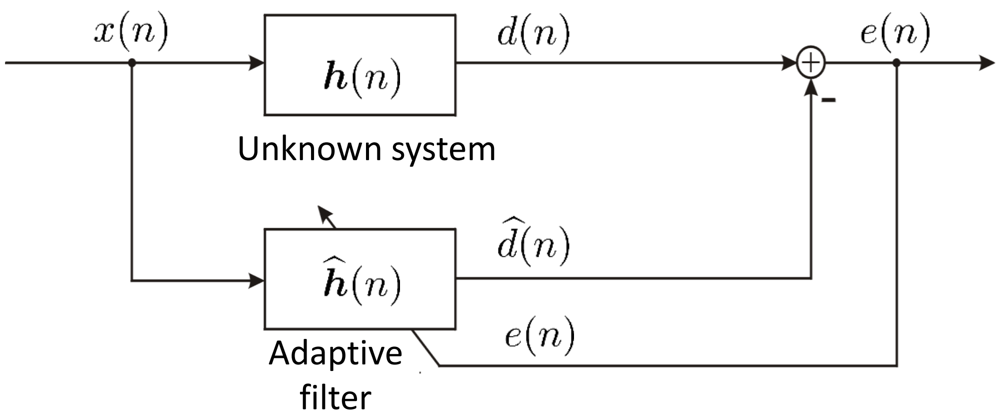
Structure:



Basics – Part 2

Output signal of the adaptive filter:

$$\begin{aligned}\widehat{d}(n) &= \sum_{i=0}^{N-1} \widehat{h}_i(n)\, x(n-i) \\ &= \widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\, \boldsymbol{x}(n) \;=\; \boldsymbol{x}^{\mathrm{T}}(n)\, \widehat{\boldsymbol{h}}(n)\end{aligned}$$



Error signal:

$$\begin{aligned}e(n) &= d(n) - \widehat{d}(n) \\ &= d(n) - \widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\, \boldsymbol{x}(n) \;=\; d(n) - \boldsymbol{x}^{\mathrm{T}}(n)\, \widehat{\boldsymbol{h}}(n)\end{aligned}$$

Mean square error:

$$\overline{e^2(n)} = \mathrm{E}\Big\{\big[d(n) - \widehat{d}(n)\big]^2\Big\} = \mathrm{E}\Big\{\big[d(n) - \widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\, \boldsymbol{x}(n)\big]^2\Big\}$$

Basics – Part 3

Mean square error:

$$\overline{e^2(n)} = \mathrm{E}\Big\{\big[d(n) - \widehat{d}(n)\big]^2\Big\} = \mathrm{E}\Big\{\big[d(n) - \widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\,\boldsymbol{x}(n)\big]^2\Big\}$$

The filter coefficients are adjusted optimally in case of orthogonality:

$$\mathrm{E}\Big\{\boldsymbol{x}(n)\,e_{\min}(n)\Big\} = \mathrm{E}\Big\{\boldsymbol{x}(n)\,\big[d(n) - \widehat{\boldsymbol{h}}_{\mathrm{opt}}^{\mathrm{T}}\,\boldsymbol{x}(n)\big]\Big\} = \mathbf{0}$$

Abbreviations:

$$\boldsymbol{R}_{xx} = \mathrm{E}\Big\{\boldsymbol{x}(n)\,\boldsymbol{x}^{\mathrm{T}}(n)\Big\}$$ (auto correlation matrix)

$$\boldsymbol{r}_{xd}(0) = \boldsymbol{r}_{xd} = \mathrm{E}\big\{\boldsymbol{x}(n)\,d(n)\big\}$$ (cross correlation vector)

Solution (according to Wiener):

$$\boldsymbol{R}_{xx}\,\widehat{\boldsymbol{h}}_{\mathrm{opt}} = \boldsymbol{r}_{xd}$$

$$\boxed{\widehat{\boldsymbol{h}}_{\mathrm{opt}} = \boldsymbol{R}_{xx}^{-1}\,\boldsymbol{r}_{xd}}$$ (assuming that the inverse exists)

Mean square error:

$$
\begin{aligned}
\overline{e^2(n)} &= \mathrm{E}\Big\{\big[d(n) - \widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\,\boldsymbol{x}(n)\big]^2\Big\} \\
&= r_{dd}(0) - 2\,\widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\,\boldsymbol{r}_{xd} + \widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\,\boldsymbol{R}_{xx}\,\widehat{\boldsymbol{h}}(n)
\end{aligned}
$$

Optimal filter vector:

$$
\widehat{\boldsymbol{h}}_{\mathrm{opt}} = \boldsymbol{R}_{xx}^{-1}\,\boldsymbol{r}_{xd}
$$

Minimum mean square error:

$$
\overline{e^2_{\mathrm{min}}(n)} = r_{dd}(0) - \boldsymbol{r}_{xd}^{\mathrm{T}}\,\widehat{\boldsymbol{h}}_{\mathrm{opt}} = r_{dd}(0) - \boldsymbol{r}_{xd}^{\mathrm{T}}\,\boldsymbol{R}_{xx}^{-1}\,\boldsymbol{r}_{xd}
$$

Basics – Part 5

Mean square error:

$$\overline{e^2(n)} = r_{dd}(0) - 2\,\widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\,\boldsymbol{r}_{xd} + \widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\,\boldsymbol{R}_{xx}\,\widehat{\boldsymbol{h}}(n)$$

Minimum mean square error :

$$\overline{e^2_{\min}(n)} = r_{dd}(0) - \widehat{\boldsymbol{h}}^{\mathrm{T}}_{\mathrm{opt}}\,\boldsymbol{r}_{xd} = r_{dd}(0) - \boldsymbol{r}^{\mathrm{T}}_{xd}\,\boldsymbol{R}^{-1}_{xx}\,\boldsymbol{r}_{xd}$$

Mean square error:

$$\begin{aligned}\overline{e^2(n)} &= r_{dd}(0) - \boldsymbol{r}^{\mathrm{T}}_{xd}\,\boldsymbol{R}^{-1}_{xx}\,\boldsymbol{r}_{xd} + \boldsymbol{r}^{\mathrm{T}}_{xd}\,\boldsymbol{R}^{-1}_{xx}\,\boldsymbol{r}_{xd}\\ &\quad -2\,\widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\,\boldsymbol{r}_{xd} + \widehat{\boldsymbol{h}}^{\mathrm{T}}(n)\,\boldsymbol{R}_{xx}\,\widehat{\boldsymbol{h}}(n)\\ &= \overline{e^2_{\min}(n)} + \underbrace{\left[\widehat{\boldsymbol{h}}(n) - \widehat{\boldsymbol{h}}_{\mathrm{opt}}\right]^{\mathrm{T}}\boldsymbol{R}_{xx}\left[\widehat{\boldsymbol{h}}(n) - \widehat{\boldsymbol{h}}_{\mathrm{opt}}\right]}\end{aligned}$$

Quadratic form ➡ unique minimum

Derivation – Part 1

Derivation with respect to the coefficients of the adaptive filter:

$$\begin{aligned}
\boldsymbol{\nabla}_{\widehat{\boldsymbol{h}}(n)} \overline{e^2(n)} &= 2\,\mathrm{E}\Big\{e(n)\,\boldsymbol{\nabla}_{\widehat{\boldsymbol{h}}(n)} e(n)\Big\} \\
&= 2\,\mathrm{E}\Big\{e(n)\,\boldsymbol{\nabla}_{\widehat{\boldsymbol{h}}(n)} \big[d(n) - \boldsymbol{x}^{\mathrm{T}}(n)\,\widehat{\boldsymbol{h}}(n)\big]\Big\} \\
&= -2\,\mathrm{E}\Big\{e(n)\,\boldsymbol{x}(n)\Big\} \qquad \longleftarrow \text{ ... needed later on ...} \\
&= -2\,\mathrm{E}\Big\{\big[d(n) - \boldsymbol{x}^{\mathrm{T}}(n)\,\widehat{\boldsymbol{h}}(n)\big]\,\boldsymbol{x}(n)\Big\} \\
&= -2\,\boldsymbol{r}_{xd} + 2\,\boldsymbol{R}_{xx}\,\widehat{\boldsymbol{h}}(n)\,.
\end{aligned}$$

Inserting $\boldsymbol{R}_{xx}\,\widehat{\boldsymbol{h}}_{\mathrm{opt}} = \boldsymbol{r}_{xd}$, results in:

$$\frac{1}{2}\,\boldsymbol{\nabla}_{\widehat{\boldsymbol{h}}(n)} \overline{e^2(n)} = \boldsymbol{R}_{xx}\left[\widehat{\boldsymbol{h}}(n) - \widehat{\boldsymbol{h}}_{\mathrm{opt}}\right].$$

Derivation – Part 2

What we have so far:

$$\frac{1}{2}\,\nabla_{\widehat{\boldsymbol{h}}(n)}\overline{e^2(n)} \;=\; \boldsymbol{R}_{xx}\left[\widehat{\boldsymbol{h}}(n)-\widehat{\boldsymbol{h}}_{\mathrm{opt}}\right].$$

Resolving it to $\widehat{\boldsymbol{h}}_{\mathrm{opt}}$ ***leads to:***

$$\widehat{\boldsymbol{h}}_{\mathrm{opt}} \;=\; \widehat{\boldsymbol{h}}(n)-\frac{1}{2}\,\boldsymbol{R}_{xx}^{-1}\,\nabla_{\widehat{\boldsymbol{h}}(n)}\overline{e^2(n)}\,.$$

With the introduction of a step size μ ***, the following adaptation rule can be formulated:***

$$\widehat{\boldsymbol{h}}(n+1) \;=\; \widehat{\boldsymbol{h}}(n)-\mu\,\boldsymbol{R}_{xx}^{-1}\,\nabla_{\widehat{\boldsymbol{h}}(n)}\overline{e^2(n)}\,.$$

Method according to Newton

Derivation – Part 3

Method according to Newton:

$$\widehat{\boldsymbol{h}}(n+1) \quad = \quad \widehat{\boldsymbol{h}}(n) - \mu\, \boldsymbol{R}_{xx}^{-1}\, \boldsymbol{\nabla}_{\widehat{\boldsymbol{h}}(n)} \overline{e^2(n)}\,.$$

Method of steepest descent:

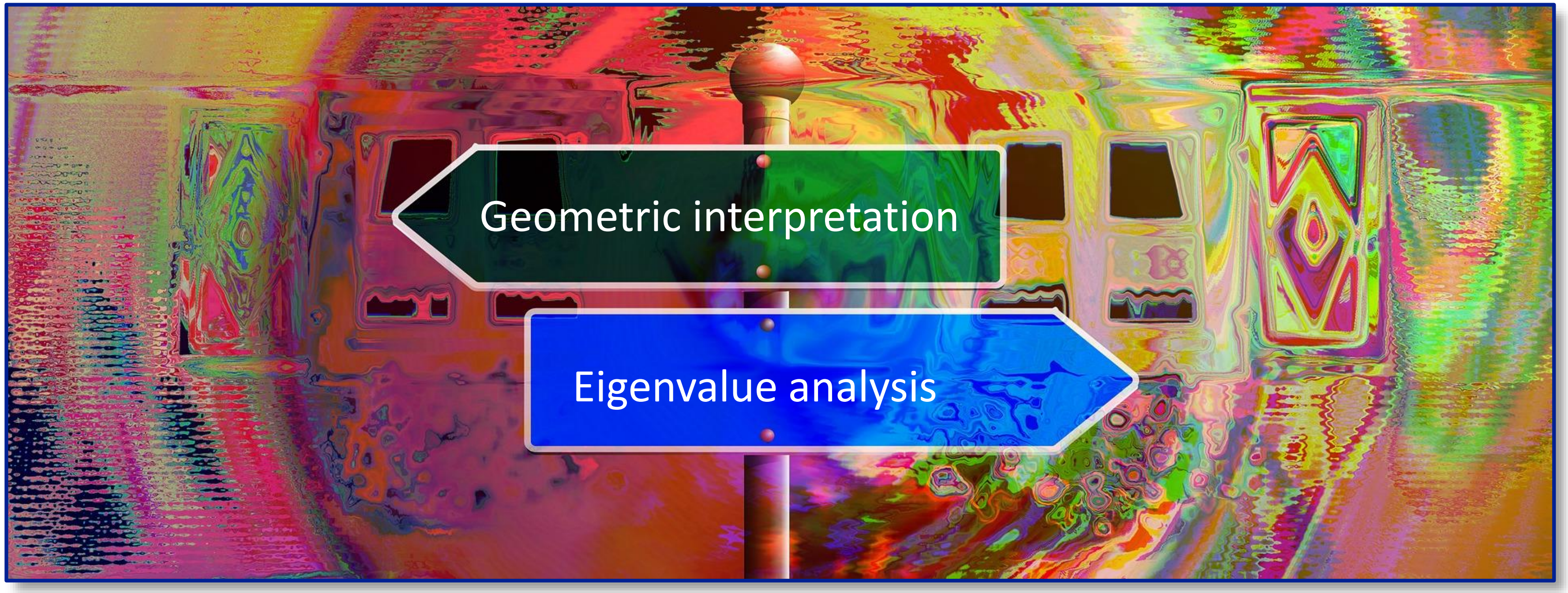
$$\begin{aligned} \widehat{\boldsymbol{h}}(n+1) \quad &= \quad \widehat{\boldsymbol{h}}(n) - \mu\, \boldsymbol{\nabla}_{\widehat{\boldsymbol{h}}(n)} \overline{e^2(n)} \\ &= \quad \widehat{\boldsymbol{h}}(n) + \mu\, \mathrm{E}\{e(n)\, \boldsymbol{x}(n)\}\,. \end{aligned}$$

For practical approaches the expectation value is replaced by its instantaneous value. This leads to the so-called least mean square algorithm:

$$\boxed{\widehat{\boldsymbol{h}}(n+1) \quad = \quad \widehat{\boldsymbol{h}}(n) + \mu\, e(n)\, \boldsymbol{x}(n)\,.}$$

LMS algorithm

Several Point-of-Views on the LMS Algorithm



Upper Bound for the Step Size

A priori error:

$$e(n|n) \;=\; e(n) \;=\; d(n) - \boldsymbol{x}^{\mathrm{T}}(n)\,\widehat{\boldsymbol{h}}(n)$$

A posteriori error:

$$\begin{aligned} e(n|n+1) &= d(n) - \boldsymbol{x}^{\mathrm{T}}(n)\,\widehat{\boldsymbol{h}}(n+1) \\ &= d(n) - \boldsymbol{x}^{\mathrm{T}}(n)\left[\widehat{\boldsymbol{h}}(n) + \mu\, e(n|n)\,\boldsymbol{x}(n)\right] \\ &= e(n|n)\,\underbrace{\left[1 - \mu\,\boldsymbol{x}^{\mathrm{T}}(n)\,\boldsymbol{x}(n)\right]}_{|\ldots| \,\le\, 1} \end{aligned}$$

Consequently:

$$0 \;<\; \mu \;<\; \frac{2}{\boldsymbol{x}^{\mathrm{T}}(n)\,\boldsymbol{x}(n)}$$

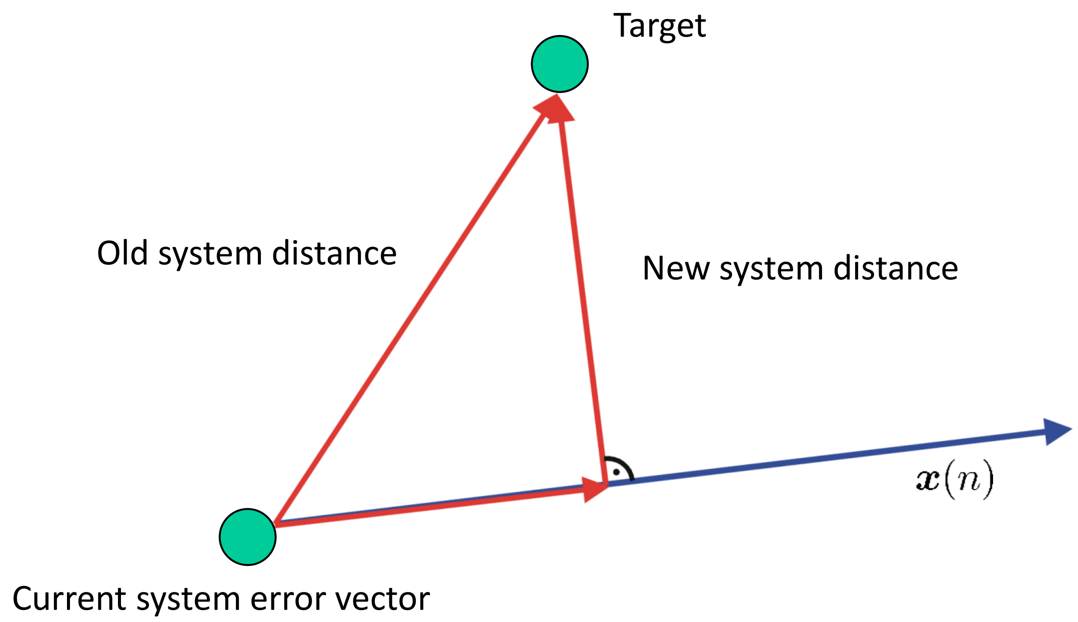
For large N and input processes with zero mean the following ***approximation*** is valid:

$$\boldsymbol{x}^{\mathrm{T}}(n)\,\boldsymbol{x}(n) \;\approx\; N\,\sigma_x^2 \quad\longrightarrow\quad \boxed{0 \;<\; \mu \;<\; \frac{2}{N\,\sigma_x^2}}$$

System Distance

How LMS adaptation changes system distance:

$$\widehat{\boldsymbol{h}}(n+1) \;=\; \widehat{\boldsymbol{h}}(n) + \mu\, e(n)\, \boldsymbol{x}(n)$$



Sign Algorithm

Update rule:

$$\widehat{\boldsymbol{h}}(n+1) \;=\; \widehat{\boldsymbol{h}}(n) + \mu\,\mathrm{sgn}\big(e(n)\big)\,\boldsymbol{x}(n)$$

with

$$\mathrm{sgn}\big(e(n)\big) \;=\; \begin{cases} 1, & \text{if } e(n) > 0, \\ 0, & \text{if } e(n) = 0, \\ -1, & \text{if } e(n) < 0. \end{cases}$$

Early algorithm with very low complexity (even used today in applications that operate at very high frequencies). It can be implemented without any multiplications (step size multiplication can be implemented as a bit shift).

Analysis of the Mean Value

Expectation of the filter coefficients:

$$\mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n+1)\Big\} \;=\; \mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n)\Big\} \;+\; \mu\,\mathrm{E}\Big\{e(n)\,\boldsymbol{x}(n)\Big\}$$

$$\mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n+1)\Big\} \;=\; \mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n)\Big\} \;=\; \widehat{\boldsymbol{h}}_\infty \quad \text{for sufficiently large } n$$

So we have ***orthogonality***:

$$\mathrm{E}\Big\{e(n)\,\boldsymbol{x}(n)\Big\} \;=\; \mathbf{0}$$

$$\mathrm{E}\Big\{\big[d(n) - \boldsymbol{x}^{\mathrm{T}}(n)\,\widehat{\boldsymbol{h}}_\infty\big]\,\boldsymbol{x}(n)\Big\} \;=\; \underbrace{\boldsymbol{r}_{xd} - \boldsymbol{R}_{xx}\,\widehat{\boldsymbol{h}}_\infty \;=\; \mathbf{0}}_{\text{Wiener solution}}$$

Convergence of the Expectations – Part 1

Into the ***equation for the LMS algorithm***

$$\widehat{\boldsymbol{h}}(n+1) \;=\; \widehat{\boldsymbol{h}}(n) + \mu\, e(n)\, \boldsymbol{x}(n)$$

we insert the equation for the ***error***

$$e(n) \;=\; d(n) - \boldsymbol{x}^{\mathrm{T}}(n)\, \widehat{\boldsymbol{h}}(n)$$

and get:

$$\begin{aligned}
\widehat{\boldsymbol{h}}(n+1) &= \widehat{\boldsymbol{h}}(n) - \mu\, \boldsymbol{x}(n)\, \boldsymbol{x}^{\mathrm{T}}(n)\, \widehat{\boldsymbol{h}}(n) + \mu\, d(n)\, \boldsymbol{x}(n) \\
&= \left[\mathbf{1} - \mu\, \boldsymbol{x}(n)\, \boldsymbol{x}^{\mathrm{T}}(n)\right] \widehat{\boldsymbol{h}}(n) + \mu\, d(n)\, \boldsymbol{x}(n)
\end{aligned}$$

Expectation of the filter coefficients:

$$\mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n+1)\Big\} \;=\; \mathrm{E}\Big\{\left[\mathbf{1} - \mu\, \boldsymbol{x}(n)\, \boldsymbol{x}^{\mathrm{T}}(n)\right] \widehat{\boldsymbol{h}}(n)\Big\} + \mu\, \mathrm{E}\Big\{d(n)\, \boldsymbol{x}(n)\Big\}$$

Convergence of the Expectations – Part 2

Expectation of the filter coefficients:

$$\mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n+1)\Big\} = \mathrm{E}\Big\{\big[\mathbf{1}-\mu\,\boldsymbol{x}(n)\,\boldsymbol{x}^{\mathrm{T}}(n)\big]\,\widehat{\boldsymbol{h}}(n)\Big\} + \mu\,\mathrm{E}\Big\{d(n)\,\boldsymbol{x}(n)\Big\}$$

Independence assumption:

$\boldsymbol{x}(n)$ and $\widehat{\boldsymbol{h}}(n)$ are statistically independent:

$$\mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n+1)\Big\} = \big(\mathbf{1}-\mu\,\boldsymbol{R}_{xx}\big)\,\mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n)\Big\} + \mu\,\boldsymbol{r}_{xd}$$

Difference between ***means and expectations***:

$$\boldsymbol{\Delta}(n) = \mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n)\Big\} - \widehat{\boldsymbol{h}}_{\infty}$$

Convergence of the means requires:

$$\lim_{n\to\infty}\boldsymbol{\Delta}(n) = \mathbf{0}$$

Convergence of the Expectations – Part 3

Recursion:

$$\mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n+1)\Big\} = \big(\mathbf{1} - \mu\,\boldsymbol{R}_{xx}\big)\,\mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n)\Big\} + \mu\,\boldsymbol{r}_{xd}$$

$$\boldsymbol{\Delta}(n) = \mathrm{E}\Big\{\widehat{\boldsymbol{h}}(n)\Big\} - \widehat{\boldsymbol{h}}_\infty$$

$$\boldsymbol{\Delta}(n+1) = \big(\mathbf{1} - \mu\,\boldsymbol{R}_{xx}\big)\,\boldsymbol{\Delta}(n) - \underbrace{\mu\,\boldsymbol{R}_{xx}\,\widehat{\boldsymbol{h}}_\infty + \mu\,\boldsymbol{r}_{xd}}$$

= 0 because of Wiener solution

Convergence requires the ***contraction of the matrix***:

$$\boxed{\boldsymbol{\Delta}(n) = \big(\mathbf{1} - \mu\,\boldsymbol{R}_{xx}\big)^n\,\boldsymbol{\Delta}(0)}$$

Convergence of the Expectations – Part 4

Convergence requires the ***contraction of the matrix*** (result from last slide):

$$\boldsymbol{\Delta}(n) \;=\; \big(\mathbf{1} - \mu\,\boldsymbol{R}_{xx}\big)^n\,\boldsymbol{\Delta}(0)$$

Case 1: ***White*** input signal

$$\boldsymbol{R}_{xx} \;=\; \sigma_x^2\,\mathbf{1}$$

$$\boldsymbol{\Delta}(n) \;=\; \big(1 - \mu\,\sigma_x^2\big)^n\,\boldsymbol{\Delta}(0)$$

Condition for the ***convergence of the mean values***:

$$\big|1 - \mu\,\sigma_x^2\big| \;<\; 1 \quad\longrightarrow\quad 0 < \mu < \frac{2}{\sigma_x^2}$$

For comparison – condition for the ***convergence of the filter coefficients***:

$$0 < \mu < \frac{2}{N\,\sigma_x^2}$$

Convergence of the Expectations – Part 5

Case 2: ***Colored*** input – assumptions

$\boldsymbol{R}_{xx}$ should be positiv definite

$$\boldsymbol{R}_{xx} = \boldsymbol{\Psi}\,\boldsymbol{\Lambda}\,\boldsymbol{\Psi}^{\mathrm{T}}$$

$\boldsymbol{\Psi}$ contains the eigenvectors of $\boldsymbol{R}_{xx}$:

$$\boldsymbol{\Psi} = \left[\boldsymbol{\psi}_0,\, \boldsymbol{\psi}_1,\, \ldots,\, \boldsymbol{\psi}_{N-1}\right]$$

$\boldsymbol{\Lambda}$ is diagonal and contains the eigenwerte λ_i of $\boldsymbol{R}_{xx}$.

The eigenvectors are pairwise different and orthonormal:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_0 & 0 & \ldots & 0 \\ 0 & \lambda_1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_{N-1} \end{bmatrix}$$

$$\boldsymbol{\psi}_i^{\mathrm{T}}\,\boldsymbol{\psi}_j = \begin{cases} 1, & \text{for } i = j, \\ 0, & \text{else.} \end{cases}$$

Convergence of the Expectations – Part 6

Putting the following results

$$\boldsymbol{R}_{xx} = \boldsymbol{\Psi}\,\boldsymbol{\Lambda}\,\boldsymbol{\Psi}^{\mathrm{T}},$$

$$\boldsymbol{\Psi} = \Big[\boldsymbol{\psi}_0,\, \boldsymbol{\psi}_1,\, \ldots,\, \boldsymbol{\psi}_{N-1}\Big],$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_0 & 0 & \ldots & 0 \\ 0 & \lambda_1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_{N-1} \end{bmatrix},$$

$$\boldsymbol{\psi}_i^{\mathrm{T}}\,\boldsymbol{\psi}_j = \begin{cases} 1, & \text{for } i = j, \\ 0, & \text{else}, \end{cases}$$

together, leads to the following notation for the ***autocorrelation matrix***:

$$\boldsymbol{R}_{xx} = \boldsymbol{\Psi}\,\boldsymbol{\Lambda}\,\boldsymbol{\Psi}^{\mathrm{T}} = \sum_{i=0}^{N-1} \lambda_i\,\boldsymbol{\psi}_i\,\boldsymbol{\psi}_i^{\mathrm{T}}.$$

Convergence of the Expectations – Part 7

Recursion:

$$\boxed{\boldsymbol{\Delta}(n) = \left(\mathbf{1} - \mu\,\boldsymbol{R}_{xx}\right)^n \boldsymbol{\Delta}(0)}$$

$$\boldsymbol{R}_{xx} = \sum_{i=0}^{N-1} \lambda_i\,\boldsymbol{\psi}_i\,\boldsymbol{\psi}_i^{\mathrm{T}}$$

$$\mathbf{1} - \mu\,\boldsymbol{R}_{xx} = \mathbf{1} - \mu \sum_{i=0}^{N-1} \lambda_i\,\boldsymbol{\psi}_i\,\boldsymbol{\psi}_i^{\mathrm{T}}$$

$$\boldsymbol{R}_{xx}\,\boldsymbol{\psi}_j = \lambda_j\,\boldsymbol{\psi}_j,$$

for $j = 0, \ldots, N-1$

$$(\mathbf{1} - \mu\,\boldsymbol{R}_{xx})\boldsymbol{\psi}_j = (1 - \mu\,\lambda_j)\,\boldsymbol{\psi}_j,$$

for $j = 0, \ldots, N-1$

$$\boldsymbol{R}_{xx} = \boldsymbol{\Psi}\,\boldsymbol{\Lambda}\,\boldsymbol{\Psi}^{\mathrm{T}}$$

$$\mathbf{1} - \mu\,\boldsymbol{R}_{xx} = \boldsymbol{\Psi}\,(\mathbf{1} - \mu\,\boldsymbol{\Lambda})\,\boldsymbol{\Psi}^{\mathrm{T}}$$

$$\left(\mathbf{1} - \mu\,\boldsymbol{R}_{xx}\right)^n = \boldsymbol{\Psi}\,\left(\mathbf{1} - \mu\,\boldsymbol{\Lambda}\right)^n\,\boldsymbol{\Psi}^{\mathrm{T}}$$

$$\boxed{\boldsymbol{\Delta}(n) = \sum_{i=0}^{N-1} \left(1 - \mu\,\lambda_i\right)^n \boldsymbol{\psi}_i\,\boldsymbol{\psi}_i^{T}\,\boldsymbol{\Delta}(0)}$$

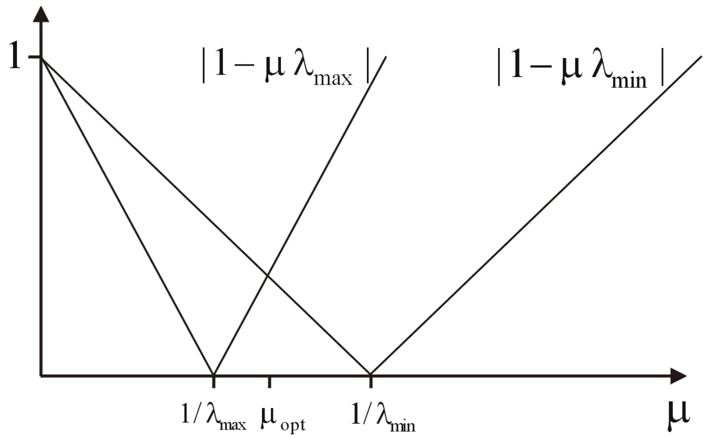
Condition for Convergence – Part 1

Previous result:

$$\boldsymbol{\Delta}(n) \;=\; \sum_{i=0}^{N-1} \left(1 - \mu\,\lambda_i\right)^n \boldsymbol{\psi}_i\,\boldsymbol{\psi}_i^T\,\boldsymbol{\Delta}(0)$$

Condition for the convergence of the ***expectations*** of the filter coefficients:

$$\left|1 - \mu\,\lambda_i\right| < 1 \quad \text{for} \quad 0 \le i \le N-1 \qquad \longrightarrow \qquad 0 \;<\; \mu \;<\; \frac{2}{\lambda_{\max}}$$



Condition for Convergence – Part 2

A (very rough) estimate for the ***largest eigenvalue***:

$$\begin{aligned} \boldsymbol{R}_{xx}\,\boldsymbol{\psi}_j &= \lambda_j\,\boldsymbol{\psi}_j \\ \boldsymbol{\psi}_j^{\mathrm{T}}\,\boldsymbol{R}_{xx}\,\boldsymbol{\psi}_j &= \lambda_j \end{aligned}$$

To all elements of the autocorrelation matrix applies: $r_{xx}(i-l) \leq \sigma_x^2$

$$\boldsymbol{\psi}_i^{\mathrm{T}}\,\boldsymbol{\psi}_j = \begin{cases} 1, & \text{for } i=j, \\ 0, & \text{else.} \end{cases}$$

Consequently:

$$\lambda_i \leq \lambda_{\max} \leq N\,\sigma_x^2$$

$$\boxed{0 < \mu < \frac{2}{N\,\sigma_x^2}}$$

Eigenvalues and Power Spectral Density – Part 1

Relation between eigenvalues and power spectral density:

Signal vector: $$\boldsymbol{x}(n) = \Big[x(n),\, x(n-1),\, x(n-2),\, \ldots\,,\, x(n-N+1)\Big]^{\mathrm{T}}$$

Autocorrelation matrix: $$\boldsymbol{R}_{xx} = \mathrm{E}\Big\{\boldsymbol{x}(n)\,\boldsymbol{x}^{\mathrm{T}}(n)\Big\} = \big[r_{xx}(k-i)\big]$$

Fourier transform: $$r_{xx}(k-i) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} S_{xx}(\Omega)\, e^{j(k-i)\Omega}\, d\Omega$$

Equation for eigenvalues: $$\boldsymbol{R}_{xx}\,\boldsymbol{\psi}_l = \lambda_l\,\boldsymbol{\psi}_l \quad \text{for } l = 0, \ldots, N-1 \qquad \boldsymbol{\psi}_i^T\,\boldsymbol{\psi}_j = \begin{cases} 1, & \text{for } i = j \\ 0, & \text{else.} \end{cases}$$

Eigenvalue: $$\boldsymbol{\psi}_l^{\mathrm{T}}\,\boldsymbol{R}_{xx}\,\boldsymbol{\psi}_l = \lambda_l = \sum_{k=0}^{N-1}\sum_{i=0}^{N-1} \psi_{lk}\,\psi_{li}\,r_{xx}(k-i)$$

$$= \sum_{k=0}^{N-1}\sum_{i=0}^{N-1} \psi_{lk}\,\psi_{li}\,\frac{1}{2\pi}\int_{-\pi}^{+\pi} S_{xx}(\Omega)\, e^{j(k-i)\Omega}\, d\Omega$$

Eigenvalues and Power Spectral Density – Part 2

Computing lower and upper bounds for the eigenvalues – part 1:

… previous result …

$$\lambda_l = \sum_{k=0}^{N-1} \sum_{i=0}^{N-1} \psi_{lk}\, \psi_{li}\, \frac{1}{2\pi} \int_{-\pi}^{+\pi} S_{xx}(\Omega)\, e^{j(k-i)\Omega}\, d\Omega$$

… exchanging the order of the sums and the integral and splitting the exponential term …

$$\lambda_l = \frac{1}{2\pi} \int_{-\pi}^{+\pi} S_{xx}(\Omega) \left| \sum_{k=0}^{N-1} \psi_{lk}\, e^{jk\Omega} \right|^2 d\Omega$$

… lower bound …

$$\lambda_l \leq S_{\max}\, \frac{1}{2\pi} \int_{-\pi}^{+\pi} \left| \sum_{k=0}^{N-1} \psi_{lk}\, e^{jk\Omega} \right|^2 d\Omega$$

… upper bound …

$$\lambda_l \geq S_{\min}\, \frac{1}{2\pi} \int_{-\pi}^{+\pi} \left| \sum_{k=0}^{N-1} \psi_{lk}\, e^{jk\Omega} \right|^2 d\Omega$$

Computing lower and upper bounds for the eigenvalues – part 2:

… exchanging again the order of the sums and the integral …

$$\frac{1}{2\pi}\int\limits_{-\pi}^{+\pi}\left|\sum_{k=0}^{N-1}\psi_{lk}\,e^{jk\Omega}\right|^2 d\Omega \;=\; \sum_{k=0}^{N-1}\sum_{i=0}^{N-1}\psi_{lk}\,\psi_{li}\,\frac{1}{2\pi}\int\limits_{-\pi}^{+\pi}e^{j(k-i)\Omega}\,d\Omega$$

… solving the integral first …

$$\frac{1}{2\pi}\int\limits_{-\pi}^{+\pi}e^{j(k-i)\Omega}\,d\Omega \;=\; \begin{cases}1, & \text{for } k=i,\\ 0, & \text{else.}\end{cases}$$

… inserting the result und using the orthonormality properties of eigenvectors …

$$\begin{aligned}\frac{1}{2\pi}\int\limits_{-\pi}^{+\pi}\sum_{k=0}^{N-1}\sum_{i=0}^{N-1}\psi_{lk}\,\psi_{li}\,e^{j(k-i)\Omega}\,d\Omega \;&=\; \sum_{k=0}^{N-1}\sum_{i=0}^{N-1}\psi_{lk}\,\psi_{li}\,\frac{1}{2\pi}\int\limits_{-\pi}^{+\pi}e^{j(k-i)\Omega}\,d\Omega\\ &=\; \sum_{k=0}^{N-1}\psi_{lk}^2 \;=\; 1\end{aligned}$$

Eigenvalues and Power Spectral Density – Part 4

Computing lower and upper bounds for the eigenvalues – part 2:

… exchanging again the order of the sums and the integral …

$$\frac{1}{2\pi} \int\limits_{-\pi}^{+\pi} \left| \sum_{k=0}^{N-1} \psi_{lk} \, e^{jk\Omega} \right|^2 d\Omega \;=\; 1$$

… inserting the result from above to obtain the upper bound …

$$\lambda_l \;\leq\; S_{\max} \frac{1}{2\pi} \int\limits_{-\pi}^{+\pi} \left| \sum_{k=0}^{N-1} \psi_{lk} \, e^{jk\Omega} \right|^2 d\Omega \;=\; S_{\max}$$

… inserting the result from above to obtain the lower bound …

$$\lambda_l \;\geq\; S_{\min} \frac{1}{2\pi} \int\limits_{-\pi}^{+\pi} \left| \sum_{k=0}^{N-1} \psi_{lk} \, e^{jk\Omega} \right|^2 d\Omega \;=\; S_{\min}$$

… finally we get…

$$\boxed{S_{\min} \leq \lambda_i \leq S_{\max} \qquad i = 0, \ldots, N-1}$$

Summary and Outlook

This week:

- Topics for the Talks
- Introductory Remarks
- Recursive Least Squares (RLS) Algorithm
- Least Mean Square Algorithm (LMS Algorithm) – Part 1

Next week:

- Least Mean Square Algorithm (LMS Algorithm) – Part 2
- Affine Projection Algorithm (AP Algorithm)
- Fast Affine Projection