3 Optimal and Adaptive Filtering

3.3: Adaptive Filtering

Optimal and Adaptive Filtering

1. **Wiener-Hopf filter**
   - Minimum Mean Square Error Estimation
   - The Wiener-Hopf solution
2. **Linear prediction**
   - The Wiener-Hopf filter as a predictor
   - Linear prediction for signal coding
3. **Adaptive filtering**
   - Steepest descent
   - Least Mean Square approach
4. **Applications of optimal and adaptive filtering**
   - …

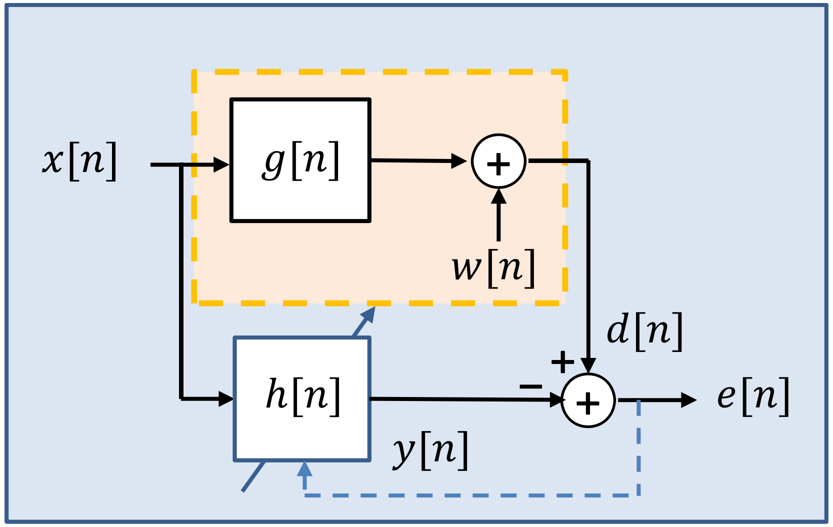
Adaptive Filtering

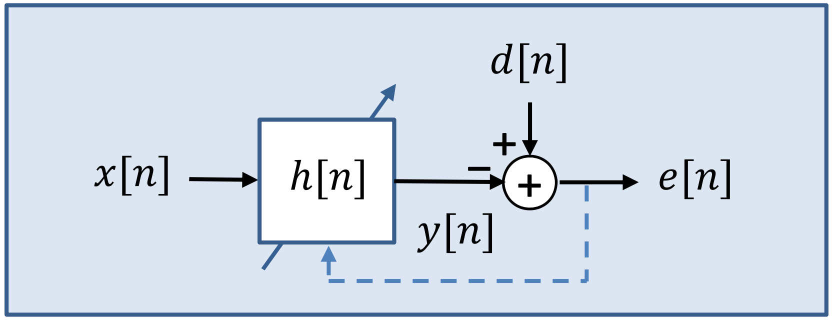
1. **Introduction**
   - Scenarios where adaptation is needed
2. **Steepest descent**
   - Study of the error performance surface
   - The minimization algorithm
   - Convergence analysis
3. **Least Mean Square approach**
   - Stochastic approximation of the gradient
   - Convergence analysis
4. **Conclusions**

Need of adaptive filtering

In the four scenarios that were presented as examples of the Wiener-Hopf filter, we can distinguish two different classes:

- **System** identification and inversion: if the system (or the system model) that is to be processed varies in time, the W-H solution has to adapt to these variations.
- **Signal** prediction and cancelation: if the processes that are analyzed are non stationary, the W-H solution has to track and adapt to their statistical variations.





$$\boxed{\underline{\mathbf{h}}_{opt} = \underline{\underline{\mathbf{R}}}_x^{-1}\underline{\mathbf{r}}_{xd}}$$

In the different scenarios, the Wiener-Hopf solution depends either on the observation or on the reference signals

Assessment of adaptive filtering

3.3

The goal of an adaptive filter is first **to find and then to track** the optimum filter as quickly and accurately as possible.

There are different algorithms for implementing the filter adaptation. Therefore, we need **criteria for assessing the quality** of these algorithms:

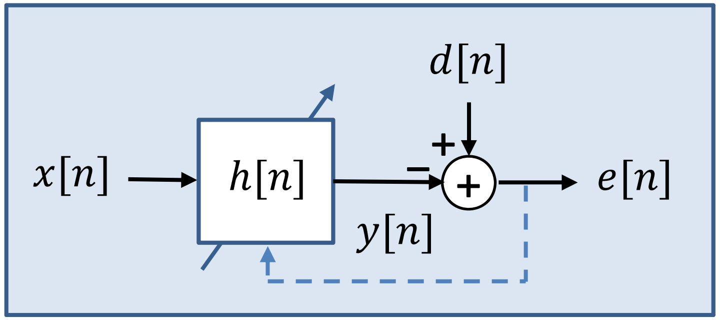
- **Speed of convergence**: (or speed of adaptation) It measures the capability of the algorithm to bring the adaptive solution to the optimal one, independently of the initial conditions. It is a transient-phase property.
- **Misadjustment**: (or quality of adaptation) It measures the stability of the reached solution, once convergence is achieved. It is due to the randomness of the input data. It is a steady-state property.
- **Tracking**: if the processes that are analyzed are non stationary, the W-H solution has to track and adapt to their statistical variations. It is a steady-state property.
- **Complexity**: Commonly, it is measured in terms of the number of operations that the algorithm requires to process a new sample, or time update. Additional concepts such as memory usage and parallelization properties can be analyzed.

Adaptive Filtering

3.3

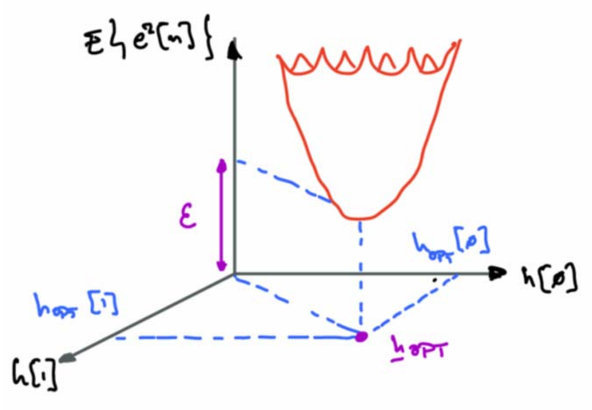
1. Introduction
    - Scenarios where adaptation is needed
2. **Steepest descent**
    - Study of the error performance surface
    - The minimization algorithm
    - Convergence analysis
3. **Least Mean Square approach**
    - Stochastic approximation of the gradient
    - Convergence analysis
4. **Conclusions**

Obtaining the W-H filter iteratively



Most adaptive filtering algorithms are obtained by simple **modifications of iterative methods** for solving deterministic optimization problems.

In the sequel, we are going to study several aspects of **gradient-based optimization techniques**, from the **theoretical viewpoint and still in a stationary scenario**, as bases for the creation and understanding of adaptive methods.



$$E\{(e[n])^2\} = \varepsilon + \left(\underline{\mathbf{h}}_{opt} - \underline{\mathbf{h}}\right)^T \underline{\underline{\mathbf{R}}}_x \left(\underline{\mathbf{h}}_{opt} - \underline{\mathbf{h}}\right)$$

To minimize the previous function, we can:

- Solve the normal equations $\underline{\mathbf{h}}_{opt} = \underline{\underline{\mathbf{R}}}_x^{-1} \underline{\mathbf{r}}_{xd}$
- **Find the minimum using an iterative solution**

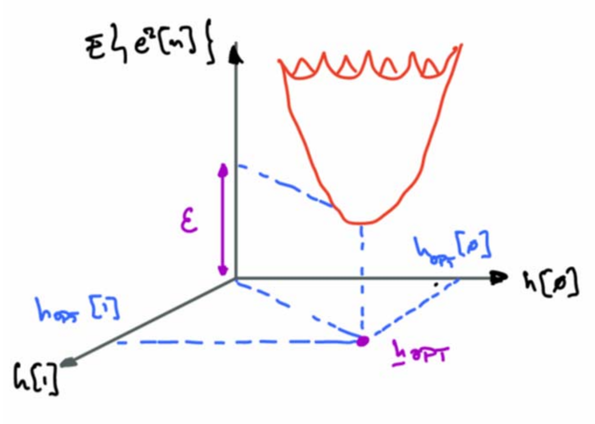
Study of the error performance surface

3.3

The Wiener-Hopf filter is optimal in the sense that it **minimizes the MSE of the prediction**; that is, the variance (power) of $e[n]$.

- For any filter, the MSE can be expressed as:

$$\boxed{E\{(e[n])^2\} = \varepsilon + \left(\underline{\mathbf{h}}_{opt} - \underline{\mathbf{h}}\right)^T \underline{\underline{\mathbf{R}}}_x \left(\underline{\mathbf{h}}_{opt} - \underline{\mathbf{h}}\right)}$$



The **error performance surface** is a **quadratic function of the filter coefficients** and represents an N-dimensional surface.

- ❑ Let us analyze the case for $N = 2$; that is $\underline{\mathbf{h}}^T = [h_0, h_1]$

$$\underline{\mathbf{h}}_{opt} - \underline{\mathbf{h}} = \underline{\Delta\mathbf{h}} = \begin{bmatrix} \Delta h_0 \\ \Delta h_1 \end{bmatrix}$$

$$\underline{\underline{\mathbf{R}}}_x = [x[n] \in \mathbb{R}] = \begin{bmatrix} r_x[0] & r_x[1] \\ r_x[1] & r_x[0] \end{bmatrix}$$

$$\Rightarrow \quad \boxed{E\{(e[n])^2\} = \varepsilon + \underline{\Delta\mathbf{h}}^T \, \underline{\underline{\mathbf{R}}}_x \, \underline{\Delta\mathbf{h}}}$$

Error performance surface

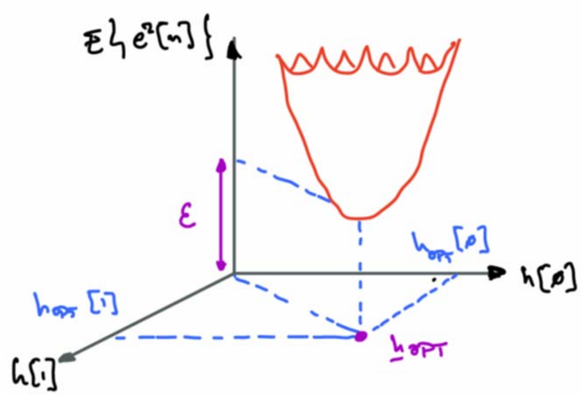
3.3

$$\boxed{E\{(e[n])^2\} = \varepsilon + \underline{\Delta\mathbf{h}}^T\, \underline{\underline{\mathbf{R}}}_x\, \underline{\Delta\mathbf{h}}}$$

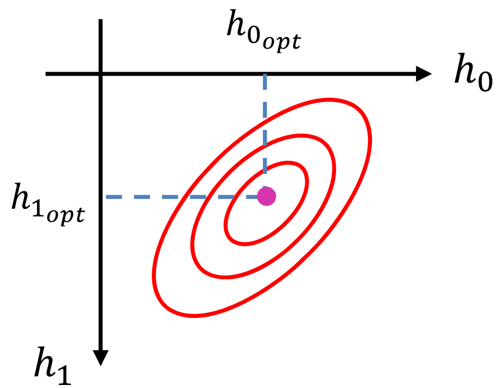
$$E\{(e[n])^2\} = \varepsilon + [\Delta h_0 \quad \Delta h_1] \begin{bmatrix} r_x[0] & r_x[1] \\ r_x[1] & r_x[0] \end{bmatrix} \begin{bmatrix} \Delta h_0 \\ \Delta h_1 \end{bmatrix}$$

$$E\{(e[n])^2\} = \varepsilon + [\Delta h_0 \quad \Delta h_1] \begin{bmatrix} r_x[0]\Delta h_0 + r_x[1]\Delta h_1 \\ \\ r_x[1]\Delta h_0 + r_x[1]\Delta h_0 \end{bmatrix} =$$

$$= \varepsilon + r_x[0](\Delta h_0)^2 + 2\Delta h_0 \Delta h_1 r_x[1] + r_x[0](\Delta h_1)^2$$



$$\boxed{\begin{aligned} E\{(e[n])^2\} = {} & \varepsilon + r_x[0]\left(h_{0_{opt}} - h_0\right)^2 + \\ & + 2r_x[1]\left(h_{0_{opt}} - h_0\right)\left(h_{1_{opt}} - h_1\right) + \\ & + r_x[0]\left(h_{1_{opt}} - h_1\right)^2 \end{aligned}}$$



Error performance surface

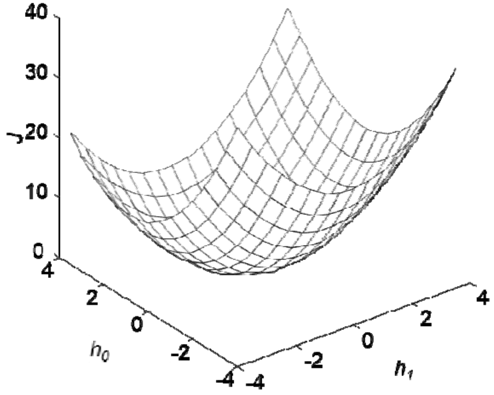
3.3

$$E\{(e[n])^2\} = \varepsilon + r_x[0]\left(h_{0_{opt}} - h_0\right)^2 + 2r_x[1]\left(h_{0_{opt}} - h_0\right)\left(h_{1_{opt}} - h_1\right) + r_x[0]\left(h_{1_{opt}} - h_1\right)^2$$

- **Example 1**: An observation signal ($x[n]$) with **low correlation** between consecutive samples:

$$\underline{\underline{\mathbf{R}}}_x = \begin{bmatrix} 1.1 & 0.1 \\ 0.1 & 1.1 \end{bmatrix} \qquad \underline{\mathbf{r}}_{xd} = \begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix}$$

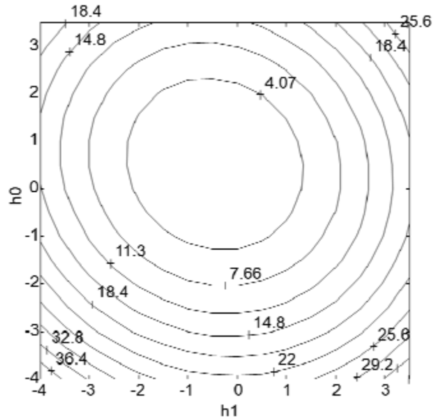
$$r_d[0] = 0.9486$$



- Given these data, we can compute:

$$\underline{\mathbf{h}}_{opt} = \underline{\underline{\mathbf{R}}}_x^{-1}\underline{\mathbf{r}}_{xd} \qquad \underline{\mathbf{h}}_{opt} = \begin{bmatrix} 0.5204 \\ -0.4526 \end{bmatrix}$$

$$\varepsilon = r_d[0] - \underline{\mathbf{h}}_{opt}^T\,\underline{\mathbf{r}}_{xd} \qquad \varepsilon = 0.4725$$



Error performance surface

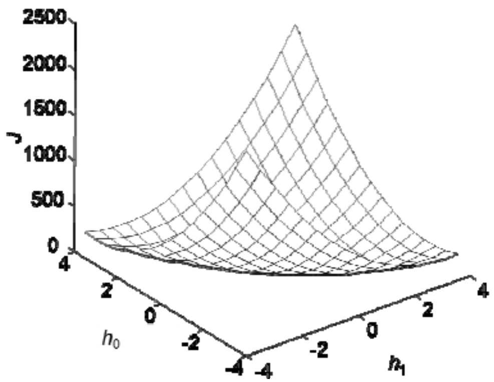
3.3

$$E\{(e[n])^2\} = \varepsilon + r_x[0]\left(h_{0_{opt}} - h_0\right)^2 + 2r_x[1]\left(h_{0_{opt}} - h_0\right)\left(h_{1_{opt}} - h_1\right) + r_x[0]\left(h_{1_{opt}} - h_1\right)^2$$

- **Example 2**: An observation signal ($x[n]$) with **high correlation** between consecutive samples:

$$\underline{\underline{\mathbf{R}}}_x = \begin{bmatrix} 40 & 39 \\ 39 & 40 \end{bmatrix} \qquad \underline{\mathbf{r}}_{xd} = \begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix}$$

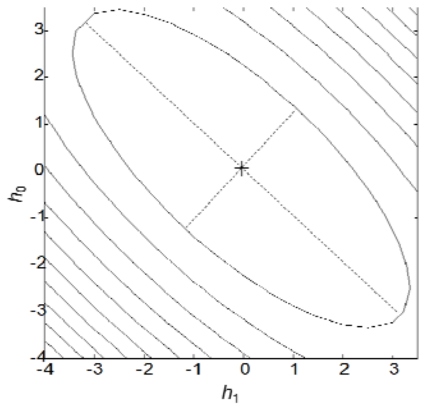
$$r_d[0] = 0.9486$$



- Given these data, we can compute:

$$\underline{\mathbf{h}}_{opt} = \underline{\underline{\mathbf{R}}}_x^{-1}\underline{\mathbf{r}}_{xd} \qquad \underline{\mathbf{h}}_{opt} = \begin{bmatrix} 0.487 \\ -0.486 \end{bmatrix}$$

$$\varepsilon = r_d[0] - \underline{\mathbf{h}}_{opt}^T\,\underline{\mathbf{r}}_{xd} \qquad \varepsilon = 0.5153$$



Adaptive Filtering

1. **Introduction**
   - Scenarios where adaptation is needed
2. **Steepest descent**
   - Study of the error performance surface
   - The minimization algorithm
   - Convergence analysis
3. **Least Mean Square approach**
   - Stochastic approximation of the gradient
   - Convergence analysis
4. **Conclusions**

Steepest descent

An iterative algorithm that obtains the minimum of the error performance surface should fulfill (k: index of the iteration):

$$E\{(e[n])^2\} = \varepsilon + \left(\underline{\mathbf{h}}_{opt} - \underline{\mathbf{h}}\right)^T \underline{\underline{\mathbf{R}}}_{\boldsymbol{x}} \left(\underline{\mathbf{h}}_{opt} - \underline{\mathbf{h}}\right)$$

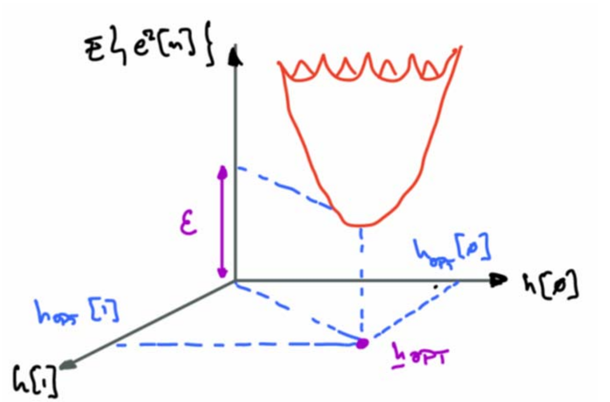
$$\lim_{k\to\infty} \underline{\mathbf{h}}^k \to \underline{\mathbf{h}}_{opt}$$

$$\lim_{k\to\infty} E\{(e[n])^2\} \to \varepsilon$$

The proposed recursion uses the information in the gradient of the function to be minimized:

$$\underline{\mathbf{h}}^{k+1} = \underline{\mathbf{h}}^k - \frac{1}{2}\mu \, \nabla_{\underline{\mathbf{h}}} \, E\{(e[n])^2\}\big|_{\underline{\mathbf{h}}^k}$$

- **Steepest descent** algorithm
- Based on the Taylor expansion around $\underline{\mathbf{h}}^k$
- The positive constant μ is known as the **step-size**
- k is a step in the **iteration**, no related to time index n

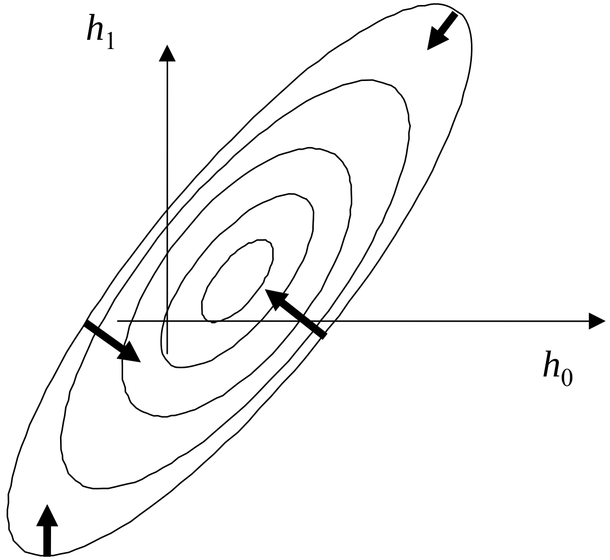


Steepest descent

The proposed recursion uses the information in the gradient of the function to be minimized:

$$\underline{\mathbf{h}}^{k+1} = \underline{\mathbf{h}}^{k} - \frac{1}{2}\mu\, \nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\}\big|_{\underline{\mathbf{h}}^k}$$

- The **step size parameter** (μ) determines the speed of convergence towards the optimum.
- The **level curves** of the error surface represent the set of point of equal MSE
- The **gradient of the error surface** adopts different directions depending on the evaluation point ($\underline{\mathbf{h}}^k$)
- The gradient of the error surface is always orthogonal to the level curves:
  - The gradient does not always aim at the optimum

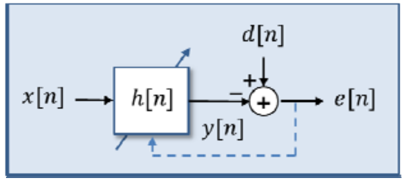


The local level curve density is related to the magnitude of the gradient at this point

Steepest descent solution of W-H

3.3

Applying this iterative algorithm to the Wiener-Hopf error performance surface, we obtain:



$$e[n] = d[n] - \underline{\mathbf{h}}^T \underline{\mathbf{x}}[n]$$

$$\underline{\mathbf{h}}^{k+1} = \underline{\mathbf{h}}^{k} - \frac{1}{2}\mu\, \nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\}|_{\underline{\mathbf{h}}^k}$$

$$\nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\} = \nabla_{\underline{\mathbf{h}}} E\left\{\left(d[n] - \underline{\mathbf{h}}^T \underline{\mathbf{x}}[n]\right)\left(d[n] - \underline{\mathbf{h}}^T \underline{\mathbf{x}}[n]\right)\right\}$$

$$\nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\} = \nabla_{\underline{\mathbf{h}}} E\left\{d[n]d[n] - d[n]\underline{\mathbf{h}}^T \underline{\mathbf{x}}[n] - \underline{\mathbf{h}}^T \underline{\mathbf{x}}[n]\, d[n] + \underline{\mathbf{h}}^T \underline{\mathbf{x}}[n]\, \underline{\mathbf{h}}^T \underline{\mathbf{x}}[n]\right\}$$

$$\nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\} = \nabla_{\underline{\mathbf{h}}} E\left\{d[n]d[n] - d[n]\underline{\mathbf{h}}^T \underline{\mathbf{x}}[n] - \underline{\mathbf{h}}^T \underline{\mathbf{x}}[n]\, d[n] + \underline{\mathbf{h}}^T \underline{\mathbf{x}}[n]\, \underline{\mathbf{x}}^T[n]\, \underline{\mathbf{h}}\right\}$$

$$\nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\} = E\left\{-2\, \underline{\mathbf{x}}[n]\, d[n] + 2\, \underline{\mathbf{x}}[n]\, \underline{\mathbf{x}}^T[n]\, \underline{\mathbf{h}}\right\} = -2\, \underline{\mathbf{r}}_{xd} + 2\underline{\underline{\mathbf{R}}}_x\, \underline{\mathbf{h}}$$

$$\boxed{\nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\}|_{\underline{\mathbf{h}}^k} = -2\, \underline{\mathbf{r}}_{xd} + 2\underline{\underline{\mathbf{R}}}_x\, \underline{\mathbf{h}}^k}$$

◀ **Gradient of the error surface**

$$\underline{\mathbf{h}}^{k+1} = \underline{\mathbf{h}}^{k} - \frac{1}{2}\mu\, \nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\}|_{\underline{\mathbf{h}}^k} \quad \Rightarrow \quad \boxed{\underline{\mathbf{h}}^{k+1} = \underline{\mathbf{h}}^{k} + \mu\left(\underline{\mathbf{r}}_{xd} - \underline{\underline{\mathbf{R}}}_x \underline{\mathbf{h}}^k\right)}$$

Adaptive Filtering

3.3

1. **Introduction**
   - Scenarios where adaptation is needed
2. **Steepest descent**
   - Study of the error performance surface
   - The minimization algorithm
   - Convergence analysis
3. **Least Mean Square approach**
   - Stochastic approximation of the gradient
   - Convergence analysis
4. **Conclusions**

Analysis of the 1D case

3.3

Let us start analyzing the **one dimension** case: $N = 1 \Longrightarrow \underline{\mathbf{h}} = h_0$

$$E\{(e[n])^2\} = \varepsilon + \left(\underline{\mathbf{h}}_{opt} - \underline{\mathbf{h}}\right)^T \underline{\underline{\mathbf{R}}}_x \left(\underline{\mathbf{h}}_{opt} - \underline{\mathbf{h}}\right)$$

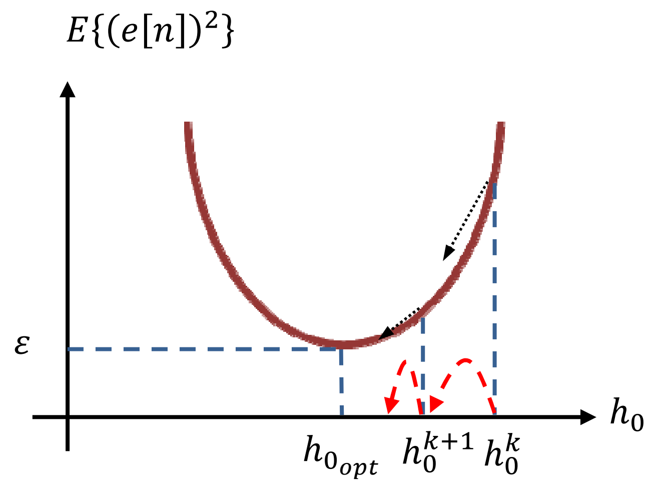
$$\underline{\underline{\mathbf{R}}}_x = r_x[0] = \lambda$$

$$\Rightarrow \quad E\{(e[n])^2\} = \varepsilon + \lambda \left(h_{0_{opt}} - h_0\right)^2$$

$$\underline{\mathbf{h}}^{k+1} = \underline{\mathbf{h}}^k - \frac{1}{2}\mu\, \nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\}|_{\underline{\mathbf{h}}^k} \quad \Rightarrow \quad h_0^{k+1} = h_0^k - \frac{1}{2}\mu\, \frac{\partial}{\partial h_0} E\{(e[n])^2\}|_{h_0^k}$$

$$h_0^{k+1} = h_0^k - \frac{1}{2}\mu\, 2\lambda \left(h_{0_{opt}} - h_0^k\right)(-1)$$

$$\boxed{h_0^{k+1} = (1 - \mu\lambda) h_0^k + \mu\lambda\, h_{0_{opt}}}$$



Convergence analysis: 1D case

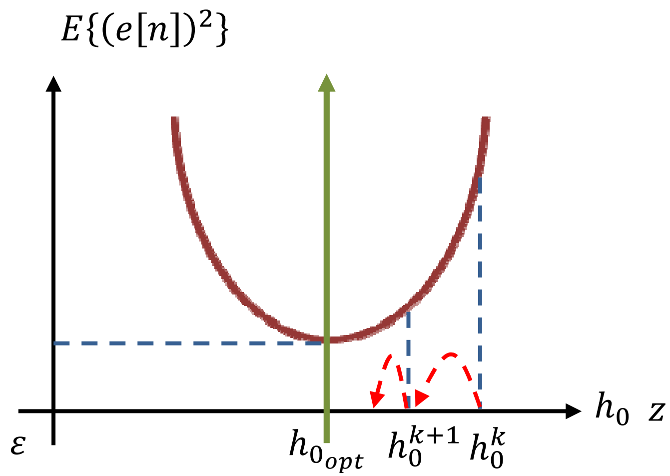
Let us take into account the specific geometry of the problem:

$$\boxed{h_0^{k+1} = (1-\mu\lambda)h_0^k + \mu\lambda\, h_{0_{opt}}}$$

$$h_0^{k+1} - h_{0_{opt}} = (1-\mu\lambda)h_0^k + \mu\lambda\, h_{0_{opt}} - h_{0_{opt}}$$

$$\underline{h_0^{k+1} - h_{0_{opt}}} = (1-\mu\lambda)\left(h_0^k - h_{0_{opt}}\right)$$

This expression allows a change of variable (z) that simplifies the convergence analysis:



$$\underline{z^{k+1}} = (1-\mu\lambda)z^k \quad \Rightarrow \quad \boxed{z^k = (1-\mu\lambda)^k z^0}$$

$$\lim_{k\to\infty} \underline{\mathbf{h}}^k \to \underline{\mathbf{h}}_{opt} \quad \Rightarrow \quad \boxed{\lim_{k\to\infty} z^k \to 0}$$

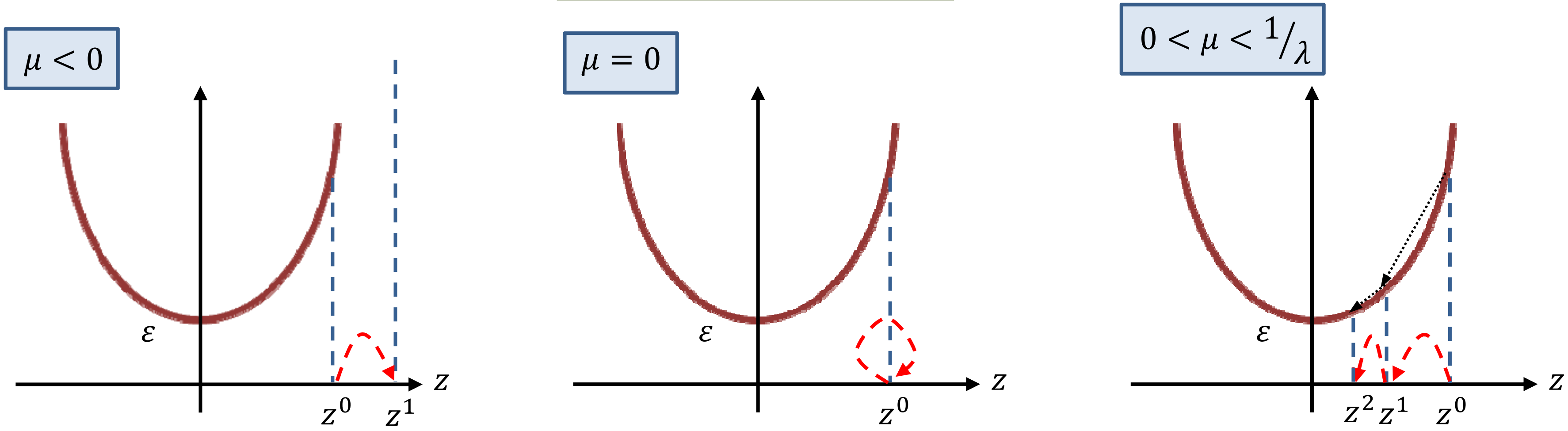
$$|1-\mu\lambda| < 1 \quad \Rightarrow \quad \left\{\begin{array}{c} 1-\mu\lambda < 1 \\ -1+\mu\lambda < 1 \end{array}\right\} \quad \Rightarrow \quad 0 < \mu\lambda < 2$$

Convergence range

$$\boxed{0 < \mu < \frac{2}{\lambda}}$$

Convergence analysis: 1D case

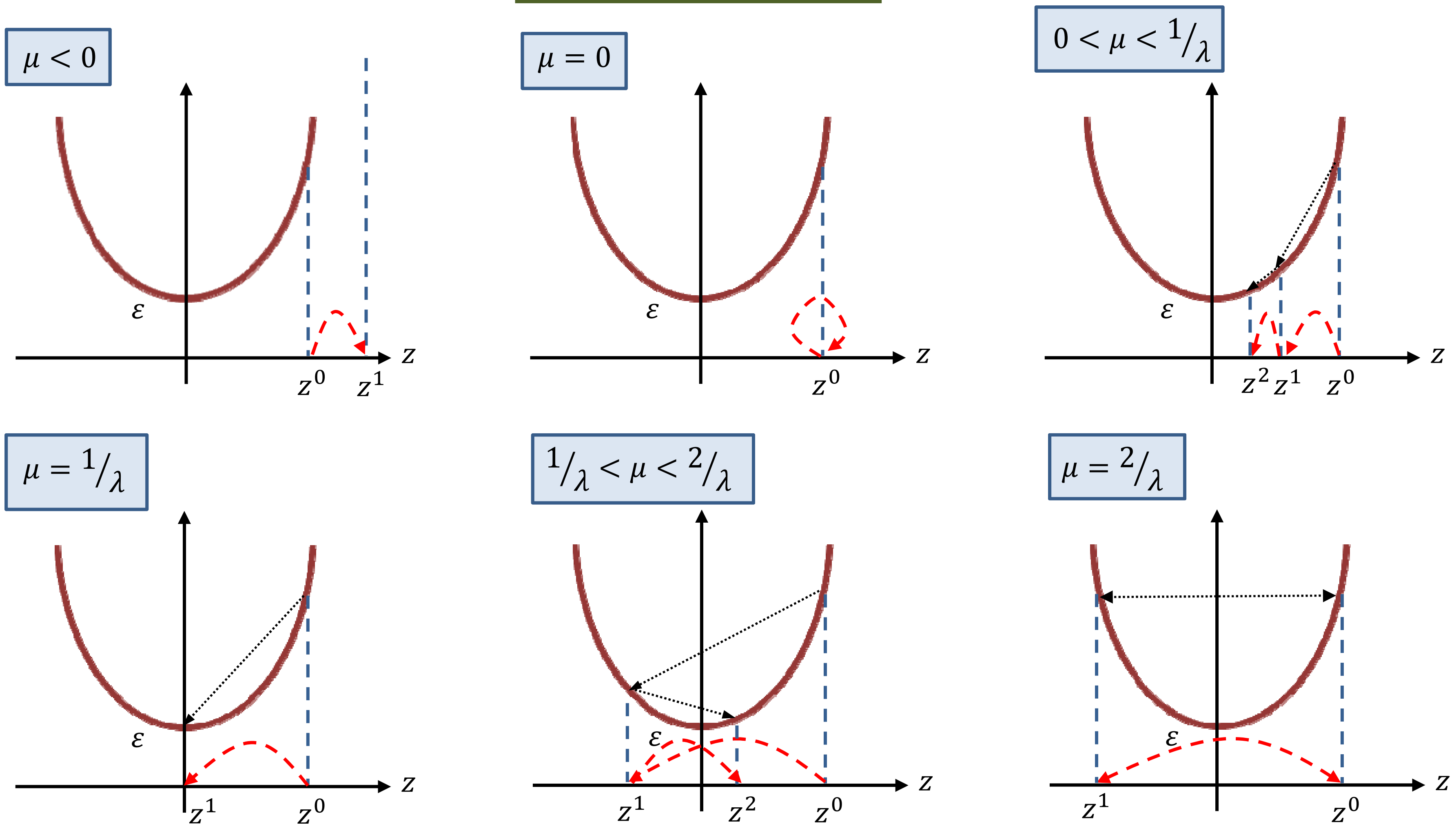
$$z^{k+1} = (1 - \mu\lambda)^k z^0$$



Convergence analysis: 1D case

3.3

$$z^{k+1} = (1 - \mu\lambda)^k z^0$$



Correlation matrix

To extend the previous analysis to the N-dimension case, we need to establish some properties of the **correlation matrix**:

- The correlation matrix is **semipositive definite**

$$\underline{\mathbf{u}}^T \underline{\underline{\mathbf{R}}}_x \underline{\mathbf{u}} \geq 0$$

$$\underline{\mathbf{u}}^T \underline{\underline{\mathbf{R}}}_x \underline{\mathbf{u}} = \underline{\mathbf{u}}^T E\{\underline{\mathbf{x}}\,\underline{\mathbf{x}}^T\}\underline{\mathbf{u}} = E\{\underline{\mathbf{u}}^T \underline{\mathbf{x}}\,\underline{\mathbf{x}}^T \underline{\mathbf{u}}\} = [\underline{\mathbf{u}}^T \underline{\mathbf{x}} = \alpha] = E\{\alpha^2\} \geq 0$$

- The **eigenvalues** of the correlation matrix are **positive**

$$\lambda_i \geq 0$$

$$\underline{\underline{\mathbf{R}}}_x \underline{\mathbf{u}} = \lambda \underline{\mathbf{u}} \quad \Rightarrow \quad \underline{\mathbf{u}}^T \underline{\underline{\mathbf{R}}}_x \underline{\mathbf{u}} = \underline{\mathbf{u}}^T \lambda \underline{\mathbf{u}} \quad \Rightarrow \quad \lambda = \frac{\underline{\mathbf{u}}^T \underline{\underline{\mathbf{R}}}_x \underline{\mathbf{u}}}{\underline{\mathbf{u}}^T \underline{\mathbf{u}}} = \frac{\alpha^2}{|\underline{\mathbf{u}}|^2} \geq \mathbf{0}$$

- It can be decomposed into an **eigenvalue and an eigenvector matrix** with $\underline{\underline{\mathbf{U}}}^T = \underline{\underline{\mathbf{U}}}^{-1}$

$$\underline{\underline{\mathbf{R}}}_x = \underline{\underline{\mathbf{U}}}\,\underline{\underline{\mathbf{\Lambda}}}\,\underline{\underline{\mathbf{U}}}^T$$

$$\underline{\underline{\mathbf{R}}}_x \underline{\mathbf{u}}_i = \lambda_i \underline{\mathbf{u}}_i \quad \Rightarrow \quad \underline{\underline{\mathbf{R}}}_x [\underline{\mathbf{u}}_1 \underline{\mathbf{u}}_2 \ \dots \underline{\mathbf{u}}_N] = [\underline{\mathbf{u}}_1 \underline{\mathbf{u}}_2 \ \dots \underline{\mathbf{u}}_N] \begin{bmatrix} \lambda_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \lambda_2 & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & \lambda_N \end{bmatrix}$$

$$\underline{\underline{\mathbf{R}}}_x \underline{\underline{\mathbf{U}}} = \underline{\underline{\mathbf{U}}}\,\underline{\underline{\mathbf{\Lambda}}} \quad \Rightarrow \quad \underline{\underline{\mathbf{R}}}_x \underline{\underline{\mathbf{U}}}\,\underline{\underline{\mathbf{U}}}^{-1} = \underline{\underline{\mathbf{U}}}\,\underline{\underline{\mathbf{\Lambda}}}\,\underline{\underline{\mathbf{U}}}^{-1} \quad \Rightarrow \quad \underline{\underline{\mathbf{R}}}_x = \underline{\underline{\mathbf{U}}}\,\underline{\underline{\mathbf{\Lambda}}}\,\underline{\underline{\mathbf{U}}}^T$$

Convergence analysis: N-dimension

We are going to perform a **change of variable** analogous to that in the 1-D case. Nevertheless, since we are in an N-D problem, we have to account for a **displacement** and a **rotation:**

$$\boxed{\underline{\mathbf{h}}^{k+1} = \underline{\mathbf{h}}^{k} + \mu\left(\underline{\mathbf{r}}_{xd} - \underline{\underline{\mathbf{R}}}_{x}\underline{\mathbf{h}}^{k}\right)}$$

$$\underline{\mathbf{h}}^{k+1} = (\underline{\underline{\mathbf{I}}} - \mu\underline{\underline{\mathbf{R}}}_{x})\underline{\mathbf{h}}^{k} + \mu\,\underline{\mathbf{r}}_{xd}$$

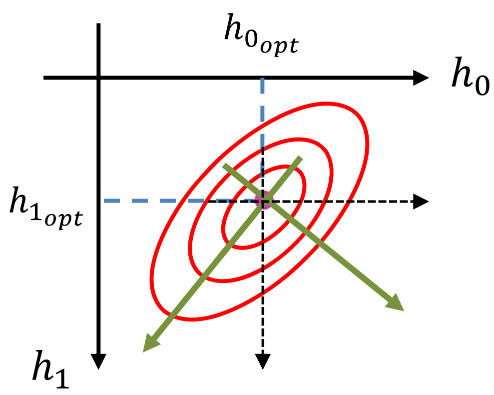
$$\underline{\mathbf{h}}^{k+1} - \underline{\mathbf{h}}_{opt} = (\underline{\underline{\mathbf{I}}} - \mu\,\underline{\underline{\mathbf{R}}}_{x})\underline{\mathbf{h}}^{k} + \mu\,\underline{\mathbf{r}}_{xd} - \underline{\mathbf{h}}_{opt}$$

$$\underline{\mathbf{r}}_{xd} = \underline{\underline{\mathbf{R}}}_{x}\underline{\mathbf{h}}_{opt}$$

$$\underline{\mathbf{h}}^{k+1} - \underline{\mathbf{h}}_{opt} = (\underline{\underline{\mathbf{I}}} - \mu\,\underline{\underline{\mathbf{R}}}_{x})\underline{\mathbf{h}}^{k} + \mu\underline{\underline{\mathbf{R}}}_{x}\underline{\mathbf{h}}_{opt} - \underline{\mathbf{h}}_{opt}$$

$$\underline{\mathbf{h}}^{k+1} - \underline{\mathbf{h}}_{opt} = (\underline{\underline{\mathbf{I}}} - \mu\,\underline{\underline{\mathbf{R}}}_{x})\underline{\mathbf{h}}^{k} - (\underline{\underline{\mathbf{I}}} - \mu\,\underline{\underline{\mathbf{R}}}_{x})\underline{\mathbf{h}}_{opt}$$

$$\boxed{\underline{\mathbf{h}}^{k+1} - \underline{\mathbf{h}}_{opt} = (\underline{\underline{\mathbf{I}}} - \mu\,\underline{\underline{\mathbf{R}}}_{x})\left(\underline{\mathbf{h}}^{k} - \underline{\mathbf{h}}_{opt}\right)}$$



Now, we can compensate for the **displacement**

Convergence analysis: N-dimension

In order to obtain the **rotation:**

$$\boxed{\underline{\mathbf{h}}^{k+1} - \underline{\mathbf{h}}_{opt} = (\underline{\underline{\mathbf{I}}} - \mu\, \underline{\underline{\mathbf{R}}}_x)(\underline{\mathbf{h}}^{k} - \underline{\mathbf{h}}_{opt})}$$

$$\boxed{\underline{\underline{\mathbf{R}}}_x = \underline{\underline{\mathbf{U}}}\,\underline{\underline{\mathbf{\Lambda}}}\,\underline{\underline{\mathbf{U}}}^T}$$

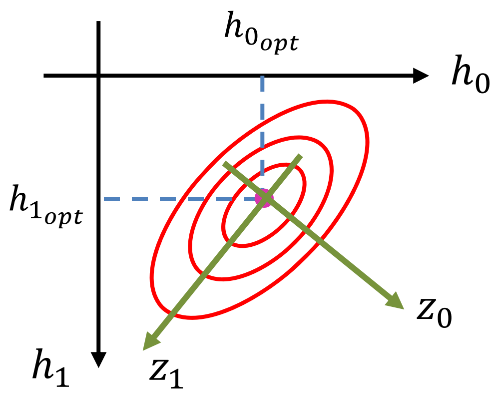
$$\underline{\mathbf{h}}^{k+1} - \underline{\mathbf{h}}_{opt} = (\underline{\underline{\mathbf{I}}} - \mu\, \underline{\underline{\mathbf{U}}}\,\underline{\underline{\mathbf{\Lambda}}}\,\underline{\underline{\mathbf{U}}}^T)(\underline{\mathbf{h}}^{k} - \underline{\mathbf{h}}_{opt})$$

$$\underline{\underline{\mathbf{U}}}^T(\underline{\mathbf{h}}^{k+1} - \underline{\mathbf{h}}_{opt}) = \underline{\underline{\mathbf{U}}}^T(\underline{\underline{\mathbf{I}}} - \mu\underline{\underline{\mathbf{U}}}\,\underline{\underline{\mathbf{\Lambda}}}\,\underline{\underline{\mathbf{U}}}^T)(\underline{\mathbf{h}}^{k} - \underline{\mathbf{h}}_{opt})$$

$$\underline{\underline{\mathbf{U}}}^T(\underline{\mathbf{h}}^{k+1} - \underline{\mathbf{h}}_{opt}) = (\underline{\underline{\mathbf{U}}}^T - \mu\underline{\underline{\mathbf{U}}}^T\underline{\underline{\mathbf{U}}}\,\underline{\underline{\mathbf{\Lambda}}}\,\underline{\underline{\mathbf{U}}}^T)(\underline{\mathbf{h}}^{k} - \underline{\mathbf{h}}_{opt})$$

$$\underline{\underline{\mathbf{U}}}^T(\underline{\mathbf{h}}^{k+1} - \underline{\mathbf{h}}_{opt}) = (\underline{\underline{\mathbf{U}}}^T - \mu\,\underline{\underline{\mathbf{\Lambda}}}\underline{\underline{\mathbf{U}}}^T)(\underline{\mathbf{h}}^{k} - \underline{\mathbf{h}}_{opt})$$

$$\underline{\underline{\mathbf{U}}}^T(\underline{\mathbf{h}}^{k+1} - \underline{\mathbf{h}}_{opt}) = (\underline{\underline{\mathbf{I}}} - \mu\,\underline{\underline{\mathbf{\Lambda}}}\,)\underline{\underline{\mathbf{U}}}^T(\underline{\mathbf{h}}^{k} - \underline{\mathbf{h}}_{opt})$$



$$\underline{\underline{\mathbf{U}}}^T(\underline{\mathbf{h}}^{k} - \underline{\mathbf{h}}_{opt}) = \underline{\mathbf{z}}^{k}$$

$$\boxed{\underline{\mathbf{z}}^{k+1} = (\underline{\underline{\mathbf{I}}} - \mu\,\underline{\underline{\mathbf{\Lambda}}}\,)\underline{\mathbf{z}}^{k}}$$

Range of Convergence: N-dimension

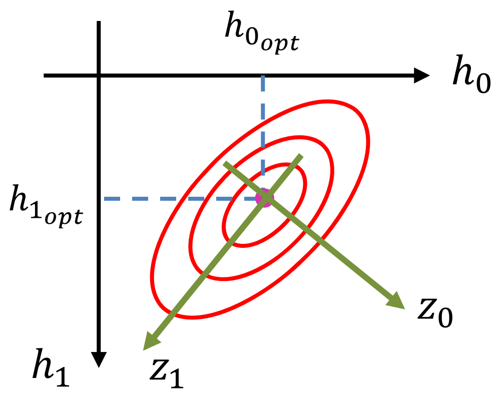
3.3

- With the previous result, we have decoupled the different dimensions of the optimization problem
- Every component can be analyzed separately:

$$\boxed{\underline{\mathbf{z}}^{k+1} = (\underline{\underline{\mathbf{I}}} - \mu\, \underline{\underline{\boldsymbol{\Lambda}}}\,)\underline{\mathbf{z}}^{k}}$$

$$z_i^{k+1} = (1-\mu\lambda_i) z_i^k \quad \Rightarrow \quad z_i^k = (1-\mu\lambda_i)^k z_i^0 \quad \Rightarrow \quad \lim_{k\to\infty} z_i^k \to \lim_{k\to\infty} (1-\mu\lambda_i)^k z_i^0$$

- At each dimension, we have: $0 < \mu < \dfrac{2}{\lambda_i}$



And, since the same μ value is used jointly in all dimensions:

$$\boxed{0 < \mu < \frac{2}{\lambda_{max}}}$$

- Do we have to compute the λ_i?
  Simpler and more **conservative policies** are used:

$$\lambda_{max} \le \sum_k \lambda_k = \mathrm{trace}(\underline{\underline{\mathbf{R}}}_{\boldsymbol{x}}) \qquad \lambda_{max} \le \sum_k r_x[0] = N r_x[0] \qquad \boxed{0 < \mu < \frac{2}{N r_x[0]}}$$

Speed of Convergence: N-dimension

3.3

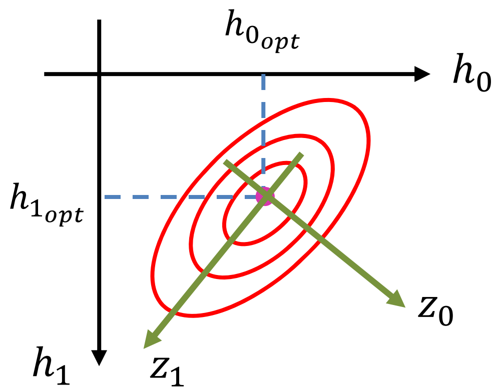
The **speed of convergence** can be quantized as the number of iteration (N_{iter}) that are necessary to reduce to a given value (ε) the distance between the achieved solution and the optimum.

In a given dimension (z_i), we can write:

$$z_i^k = (1-\mu\lambda_i)^k z_i^0 \quad \Rightarrow \quad |1-\mu\lambda_i|^{N_{iter}} = \varepsilon \quad \Rightarrow \quad N_{iter} = \frac{\ln\varepsilon}{\ln|1-\mu\lambda_i|}$$

When generalized to the N dimensions, it can be shown that for small values of μ:

$$\boxed{N_{iter} \propto -\ln\varepsilon \frac{\lambda_{max}}{\lambda_{min}}}$$

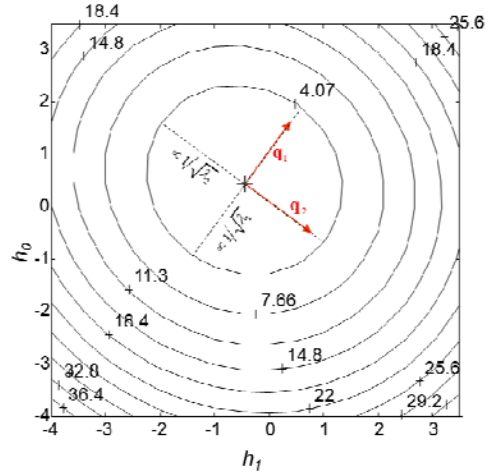


The speed of convergence is proportional to the **dispersion of the eigenvalues**.

Eigenvalue dispersion: Examples

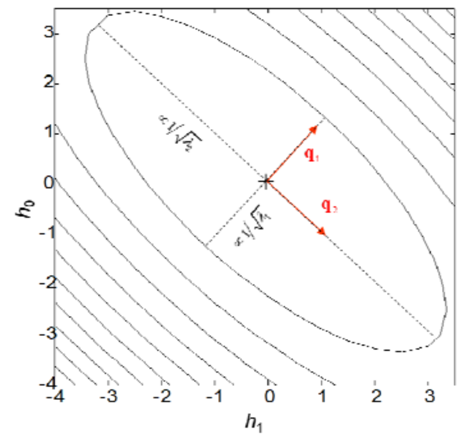
- **Example 1**: An observation signal ($x[n]$) with **low correlation** between consecutive samples: **low eigenvalue dispersion**

$$\underline{\underline{\mathbf{R}}}_x = \begin{bmatrix} 1.1 & 0.1 \\ 0.1 & 1.1 \end{bmatrix} \qquad \lambda_1 = 1.2 \qquad \lambda_2 = 1.0$$



- **Example 2**: An observation signal ($x[n]$) with **high correlation** between consecutive samples: **high eigenvalue dispersion**

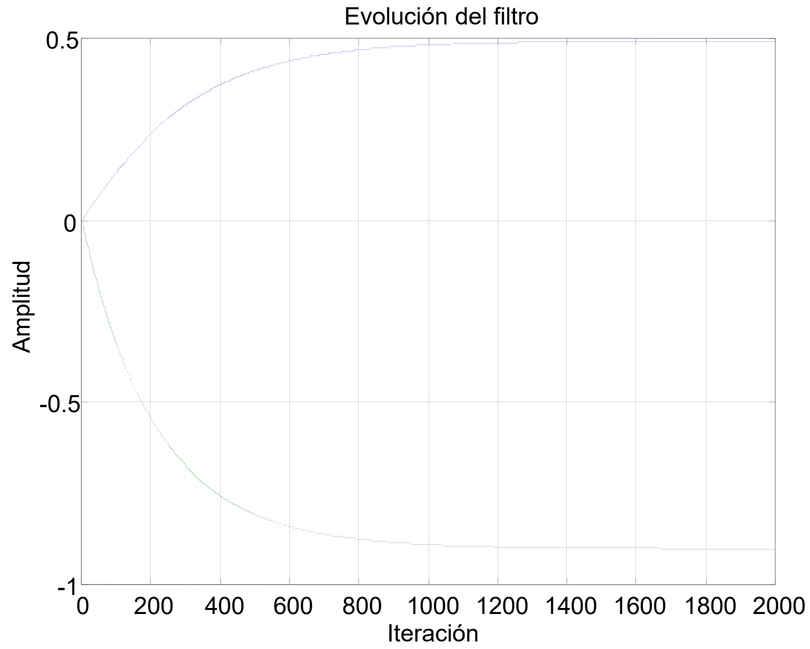
$$\underline{\underline{\mathbf{R}}}_x = \begin{bmatrix} 40 & 39 \\ 39 & 40 \end{bmatrix} \qquad \lambda_1 = 79 \qquad \lambda_2 = 1.0$$

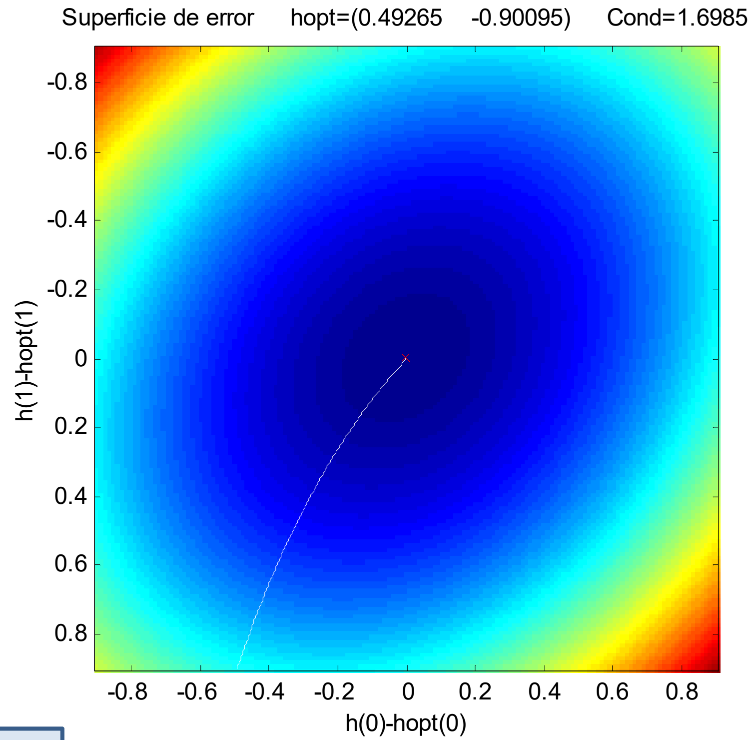


Convergence: Examples

Example 1.a: An observation signal ($x[n]$) with **low correlation** between consecutive samples: **low eigenvalue dispersion**.

$$\lambda_{max} = 7.39 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 1.71 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.5 \quad -0.9]$$





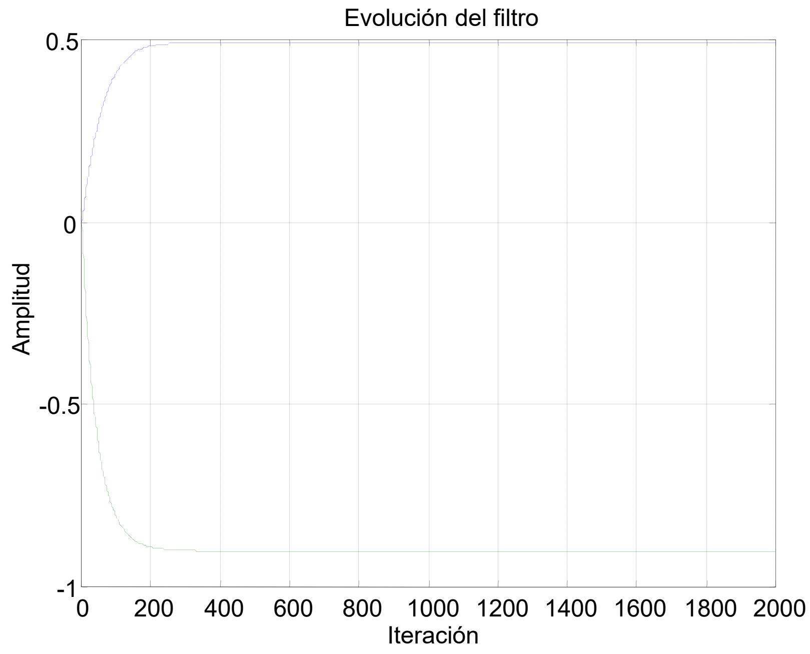
$$\boxed{\mu = 0.001}$$

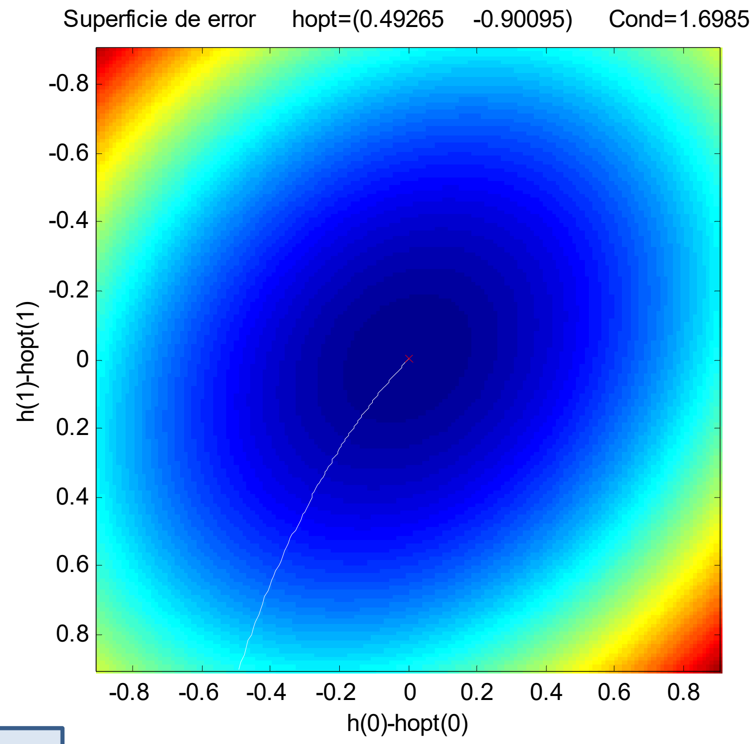
Convergence: Examples

3.3

Example 1.b: An observation signal ($x[n]$) with **low correlation** between consecutive samples: **low eigenvalue dispersion**.

$$\lambda_{max} = 7.39 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 1.71 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.5 \quad -0.9]$$



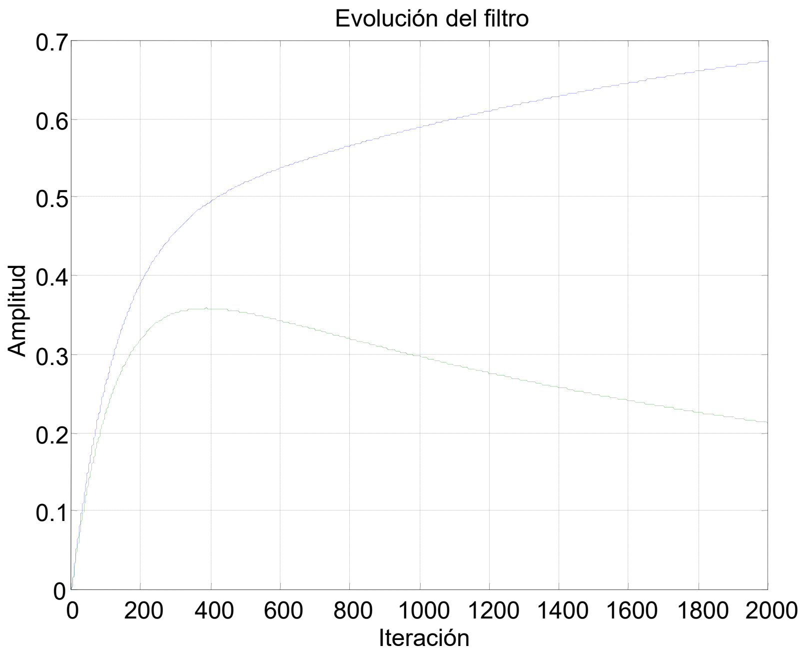


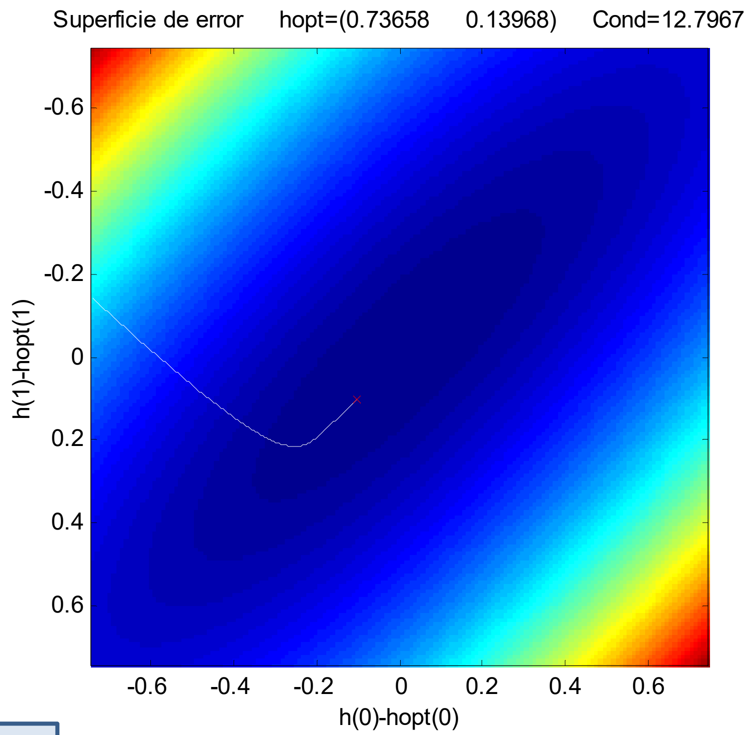
$$\boxed{\mu = 0.005}$$

Convergence: Examples

Example 2.a: An observation signal ($x[n]$) with **high correlation** between consecutive samples: **high eigenvalue dispersion**.

$$\lambda_{max} = 6.8 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 13 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.75 \quad 0.125]$$



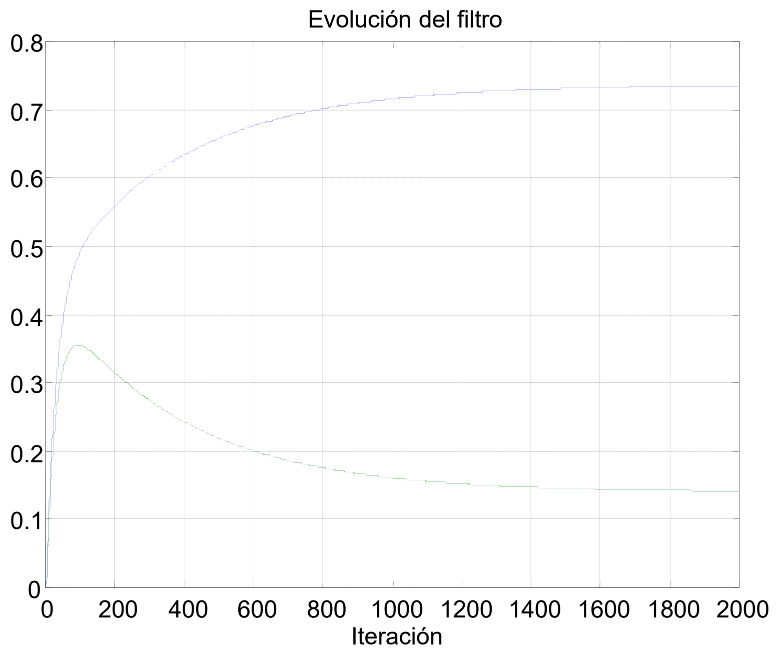


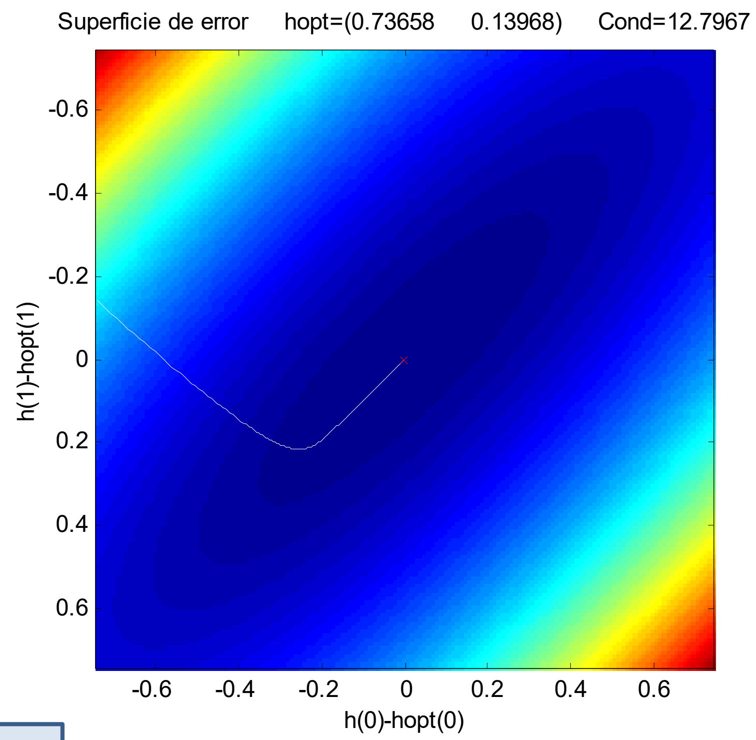
$$\boxed{\mu = 0.001}$$

Convergence: Examples

Example 2.b: An observation signal ($x[n]$) with **high correlation** between consecutive samples: **high eigenvalue dispersion**.

$$\lambda_{max} = 6.8 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 13 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.75 \quad 0.125]$$





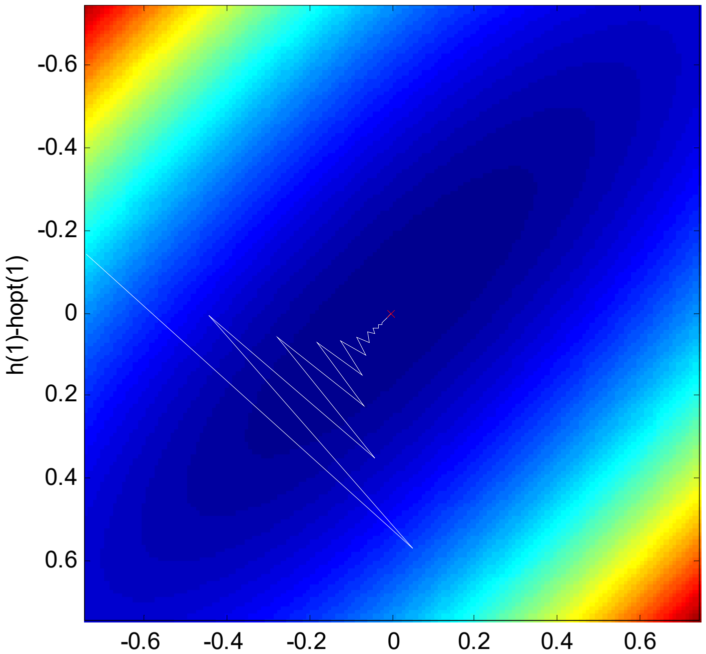
$$\boxed{\mu = 0.005}$$

Convergence: Examples

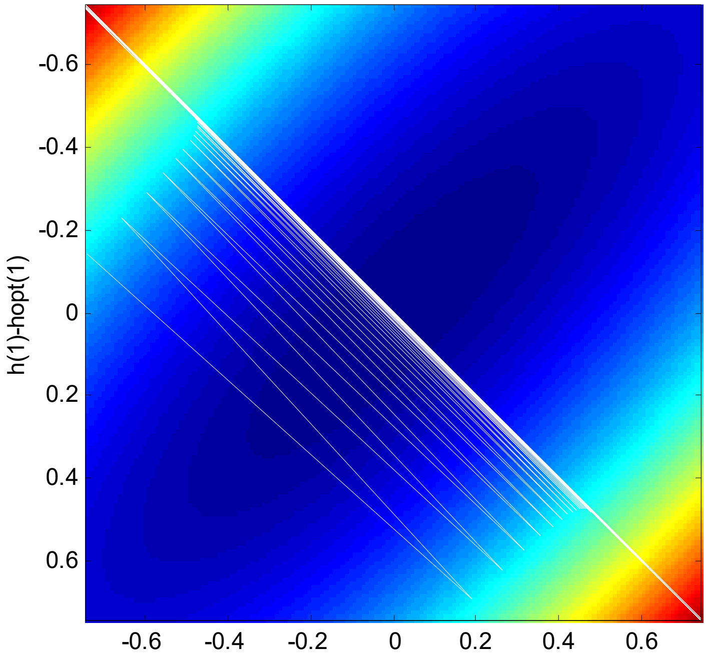
3.3

Example 2.c: An observation signal ($x[n]$) with **high correlation** between consecutive samples: **high eigenvalue dispersion**.

$$\lambda_{max} = 6.8 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 13 \qquad \frac{2}{Nr_x[0]} = 0.2714 \qquad \frac{2}{\lambda_{max}} = 0.2930$$



$\mu = 0.25$



$\mu = 0.2934$

Adaptive Filtering

3.3

1. **Introduction**
   - Scenarios where adaptation is needed
2. **Steepest descent**
   - Study of the error performance surface
   - The minimization algorithm
   - Convergence analysis
3. **Least Mean Square approach**
   - Stochastic approximation of the gradient
   - Convergence analysis
4. **Conclusions**

Stochastic approximation of the gradient

3.3

The steepest descent recursion for the Wiener-Hopf problem is:

$$\boxed{\underline{\mathbf{h}}^{k+1} = \underline{\mathbf{h}}^{k} + \mu\left(\underline{\mathbf{r}}_{xd} - \underline{\underline{\mathbf{R}}}_{x}\underline{\mathbf{h}}^{k}\right)}$$

However, in a real problem, **neither the correlation matrix nor the cross-correlation vector are known** and both have to be estimated.

In this application, as signals are assumed to be non-stationary, we cannot use an estimator with a large memory (no accumulation of previous data):

$$\underline{\underline{\mathbf{R}}}_{x} = E\left\{\underline{\mathbf{x}}[n]\,\underline{\mathbf{x}}^{T}[n]\right\} \quad \Rightarrow \quad \hat{\underline{\underline{\mathbf{R}}}}_{x}\left(\underline{\mathbf{x}}\right) = \underline{\mathbf{x}}[n]\,\underline{\mathbf{x}}^{T}[n]$$

$$\underline{\mathbf{r}}_{xd} = E\left\{\underline{\mathbf{x}}[n]d[n]\right\} \quad \Rightarrow \quad \hat{\underline{\mathbf{r}}}_{xd}\left(\underline{\mathbf{x}}, d[n]\right) = \underline{\mathbf{x}}[n]d[n]$$

Instantaneous estimates of the correlation matrix and cross-correlation vector

This way, the gradient of the error performance surface is estimated as:

$$\nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\}|_{\underline{\mathbf{h}}^k} = -2\,\underline{\mathbf{r}}_{xd} + 2\underline{\underline{\mathbf{R}}}_{x}\,\underline{\mathbf{h}}^{k} \approx \boxed{-2\underline{\mathbf{x}}[n]d[n] + 2\underline{\mathbf{x}}[n]\,\underline{\mathbf{x}}^{T}[n]\,\underline{\mathbf{h}}^{n}}$$

Stochastic approximation of the gradient

Least Mean Square approach

Using the stochastic approximation of the gradient in the recursion:

$$\nabla_{\underline{\mathbf{h}}}\, E\{(e[n])^2\}|_{\underline{\mathbf{h}}^k} \approx -2\underline{\mathbf{x}}[n]d[n] + 2\underline{\mathbf{x}}[n]\, \underline{\mathbf{x}}^T[n]\, \underline{\mathbf{h}}^n$$

$$\underline{\mathbf{h}}^{k+1} = \underline{\mathbf{h}}^k + \mu\left(\underline{\mathbf{r}}_{xd} - \underline{\underline{\mathbf{R}}}_x \underline{\mathbf{h}}^k\right) \quad \Rightarrow \quad \underline{\mathbf{h}}^{n+1} = \underline{\mathbf{h}}^n + \mu\left(\underline{\mathbf{x}}[n]d[n] - \underline{\mathbf{x}}[n]\, \underline{\mathbf{x}}^T[n]\, \underline{\mathbf{h}}^n\right)$$

$$\underline{\mathbf{h}}^{n+1} = \underline{\mathbf{h}}^n + \mu\, \underline{\mathbf{x}}[n]\left(d[n] - \underline{\mathbf{x}}^T[n]\, \underline{\mathbf{h}}^n\right) \qquad e[n] = d[n] - \underline{\mathbf{h}}^T \underline{\mathbf{x}}[n]$$

$$\boxed{\underline{\mathbf{h}}^{n+1} = \underline{\mathbf{h}}^n + \mu\, \underline{\mathbf{x}}[n]e[n]}$$

◀ **Least Mean Square (LMS) approach**

LMS algorithm:

- Filtering the signal: $y[n] = \underline{\mathbf{x}}^T[n]\underline{\mathbf{h}}^n$
- Computing the error: $e[n] = d[n] - y[n]$
- Updating the coefficients: $\underline{\mathbf{h}}^{n+1} = \underline{\mathbf{h}}^n + \mu\, \underline{\mathbf{x}}[n]e[n]$

Adaptive Filtering

3.3

1. **Introduction**
   - Scenarios where adaptation is needed
2. **Steepest descend**
   - Study of the error performance surface
   - The minimization algorithm
   - Convergence analysis
3. **Least Mean Square approach**
   - Stochastic approximation of the gradient
   - Convergence analysis
4. **Conclusions**

Convergence of the LMS

3.3

- We study the convergence of the LMS algorithm in a **stationary scenario**.
- As the gradient is estimated, the resulting value is random. Therefore, we need **to study the algorithm convergence in statistical terms**
- Study of the convergence in an **average sense**:

$$\underline{\mathbf{h}}^{n+1} = \underline{\mathbf{h}}^{n} + \mu \left(\underline{\mathbf{x}}[n] d[n] - \underline{\mathbf{x}}[n]\, \underline{\mathbf{x}}^{T}[n]\, \underline{\mathbf{h}}^{n} \right)$$

We assume that observations and coefficients are approximately independent

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

$$E\{\underline{\mathbf{h}}^{n+1}\} = E\{\underline{\mathbf{h}}^{n}\} + \mu\, E\{\underline{\mathbf{x}}[n] d[n]\} - \mu\, E\{\underline{\mathbf{x}}[n]\, \underline{\mathbf{x}}^{T}[n]\, \underline{\mathbf{h}}^{n}\}$$

$$E\{\underline{\mathbf{h}}^{n+1}\} = E\{\underline{\mathbf{h}}^{n}\} + \mu\, E\{\underline{\mathbf{x}}[n] d[n]\} - \mu\, E\{\underline{\mathbf{x}}[n]\, \underline{\mathbf{x}}^{T}[n]\} E\{\underline{\mathbf{h}}^{n}\}$$

$$\boxed{E\{\underline{\mathbf{h}}^{n+1}\} = E\{\underline{\mathbf{h}}^{n}\} + \mu \underline{\mathbf{r}}_{xd} - \mu\, \underline{\underline{\mathbf{R}}}_{x}\, E\{\underline{\mathbf{h}}^{n}\}}$$

In the average sense, we obtain **the same iteration equation** as with the Steepest Descent method

Convergence of the LMS (mean sense)

3.3

- The **step size** (μ) has to fulfill the same restrictions as in the Steepest Decent (SD) algorithm to achieve convergence:

$$\boxed{0 < \mu < \frac{2}{\lambda_{max}}}$$

- The **speed of convergence** is the same in both cases (LMS, in the mean sense, and SD):

$$\boxed{N_{iter} \propto -\ln \varepsilon \frac{\lambda_{max}}{\lambda_{min}}}$$

- As in the SD algorithm, a **conservative policy** is adopted for the step size:

$$\boxed{0 < \mu < \frac{2}{N\hat{r}_x[0]}}$$

In some cases, the dynamics of the input signal (that is, $r_x[0]$) is not constant due to non-stationarity. In such a case, the step-size should be updated to guarantee convergence. **Normalized LMS**:

- **Conservative** value of the **step size**: $$\mu = \frac{2\alpha}{N\hat{r}_x[0]} \quad \text{with} \quad 0 < \alpha < 1$$
- **Dynamic estimation** of the input power:
    - Instantaneous estimation: $$N\hat{r}_x[0] = \underline{\mathbf{x}}^T[n]\underline{\mathbf{x}}[n]$$
    - Time-averaged estimation $$\hat{r}_x[0;n] = \gamma\hat{r}_x[0;n-1] + (1-\gamma)|x[n]|^2$$

Misadjustment of the LMS

3.3

Although the LMS converges in the mean sense, the fact of estimating the gradient produces an increase in variance of minimum error achieved.

This is known as the LMS **steady-state excess MSE** (in absolute value) or as the LMS **misadjustment** (in relative terms):

$$E\{\hat{e}^2[n]\} - \varepsilon \approx \frac{\mu\varepsilon \sum_{i=1}^{N} \lambda_i}{2 - \mu \sum_{i=1}^{N} \lambda_i} = \frac{\mu N r_x[0]}{2 - \mu N r_x[0]}$$

$$M = \frac{E\{\hat{e}^2[n]\} - \varepsilon}{\varepsilon} \approx \frac{\mu N r_x[0]}{2 - \mu N r_x[0]} \quad \Rightarrow \text{if } \mu \ll \frac{2}{N r_x[0]} \Rightarrow \quad \boxed{M \approx \frac{\mu}{2} \sum_{i=1}^{N} \lambda_i = \frac{\mu}{2} N r_x[0]}$$

The μ parameter in the LMS algorithm:

- It is bounded to ensure convergence
- The speed of convergence increases with μ
- The misadjustement is proportional to μ

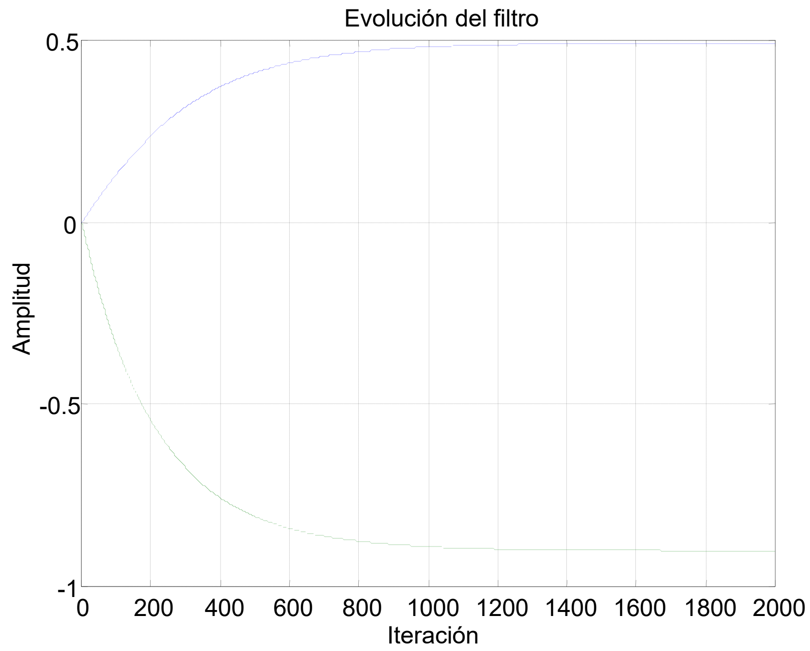
Moreover:

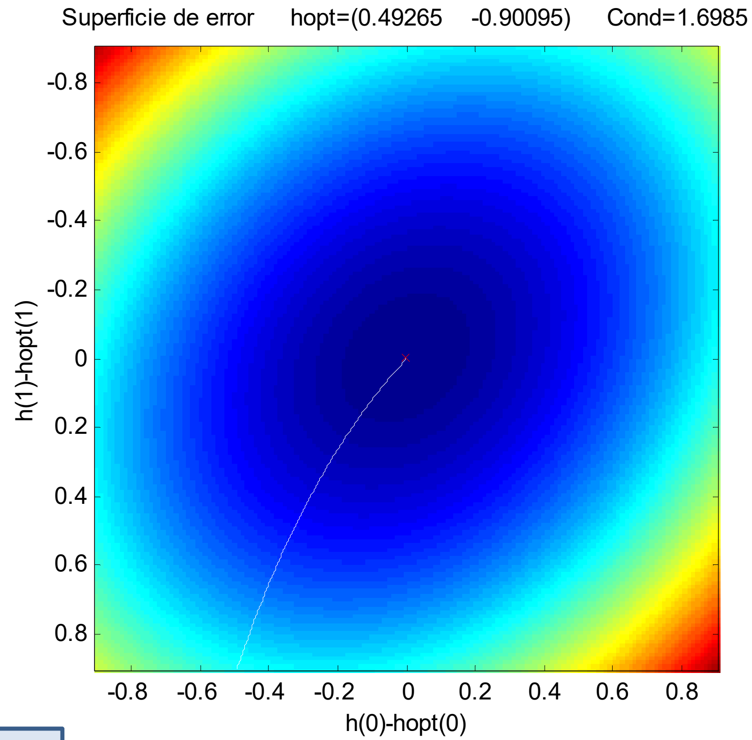
- Eigenvalue dispersion affects the speed of convergence but not the misadjustement
- Increasing the power of the signal increases the misadjustement

Convergence & Misadjustment: Examples

Example 1.a: An observation signal ($x[n]$) with **low correlation** between consecutive samples: **low eigenvalue dispersion**.

$$\lambda_{max} = 7.39 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 1.71 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.5 \quad -0.9]$$



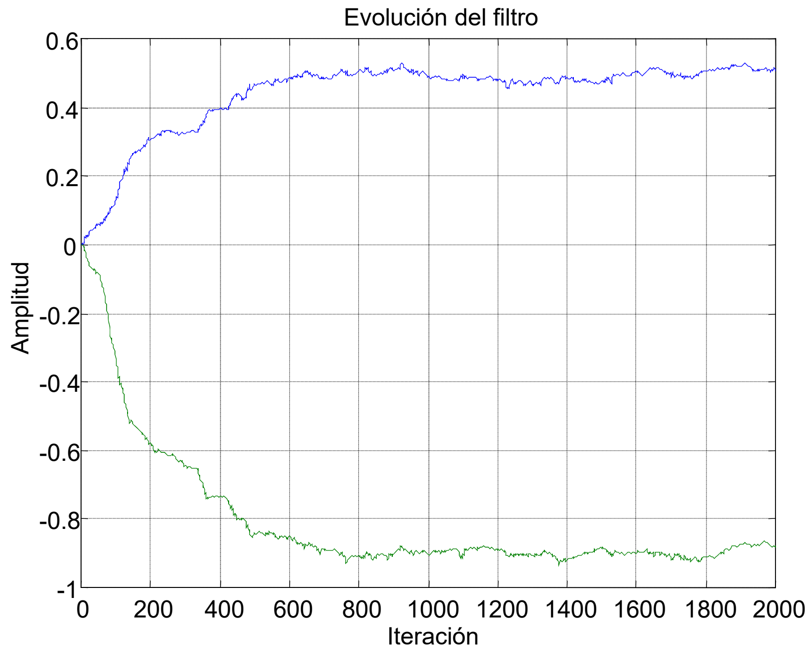


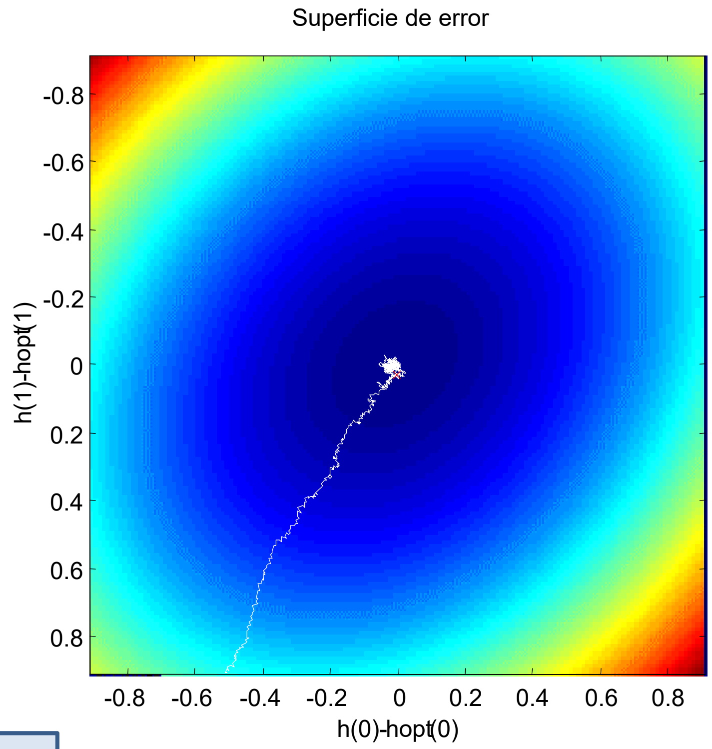
$$\boxed{\mu = 0.001}$$

Convergence & Misadjustment: Examples

Example 1.a: An observation signal ($x[n]$) with **low correlation** between consecutive samples: **low eigenvalue dispersion**.

$$\lambda_{max} = 7.39 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 1.71 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.5 \quad -0.9]$$



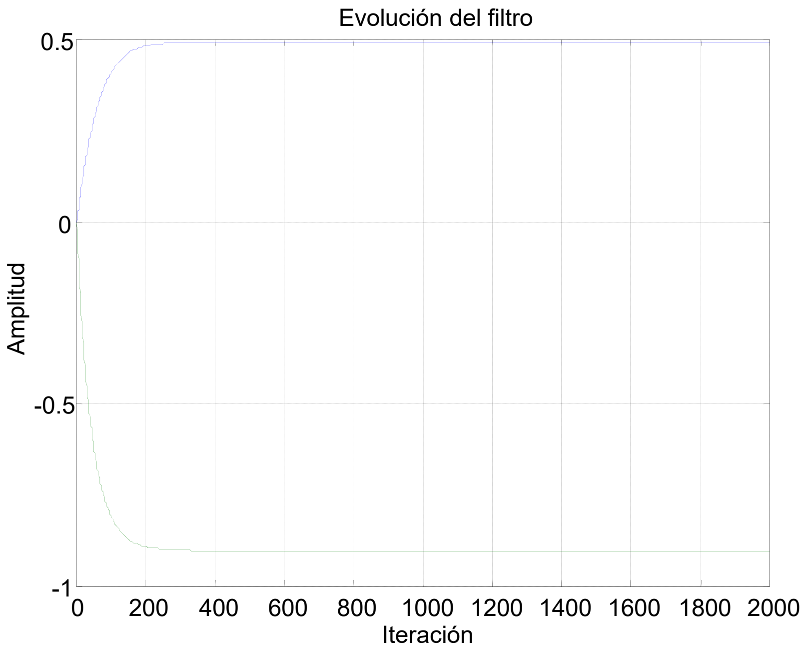


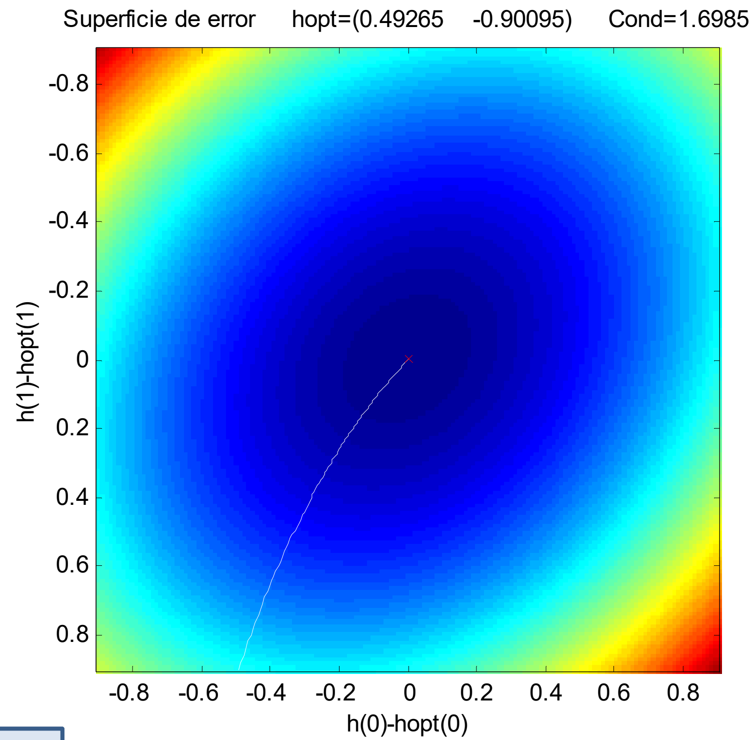
$$\boxed{\mu = 0.001}$$

Convergence & Misadjustment: Examples

Example 1.b: An observation signal ($x[n]$) with **low correlation** between consecutive samples: **low eigenvalue dispersion**.

$$\lambda_{max} = 7.39 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 1.71 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.5 \quad -0.9]$$





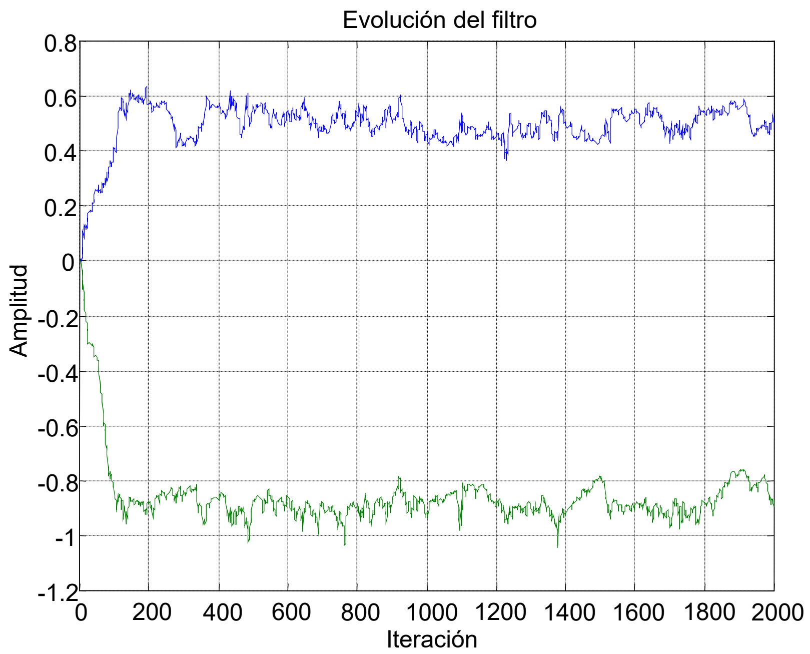
$$\boxed{\mu = 0.005}$$

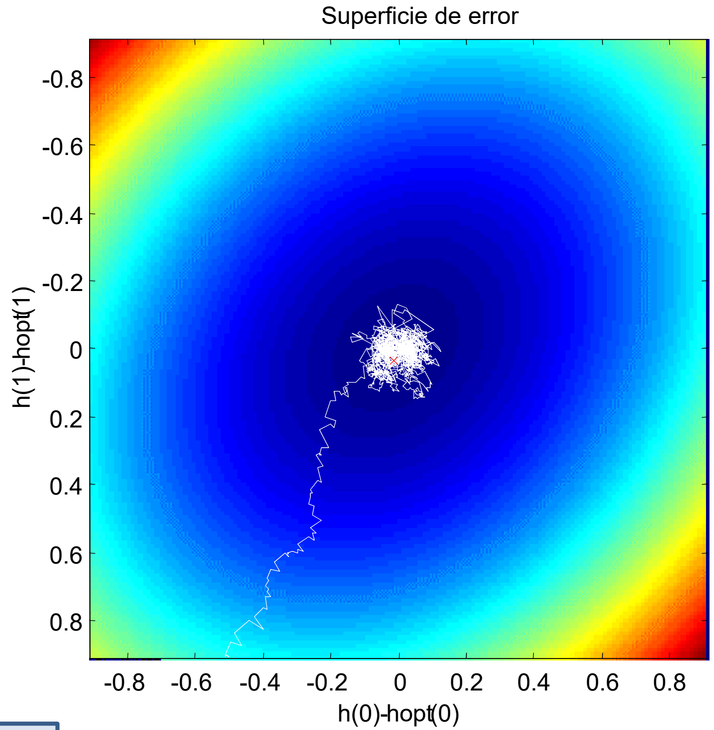
Convergence & Misadjustment: Examples

3.3

Example 1.b: An observation signal ($x[n]$) with **low correlation** between consecutive samples: **low eigenvalue dispersion**.

$$\lambda_{max} = 7.39 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 1.71 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.5 \quad -0.9]$$



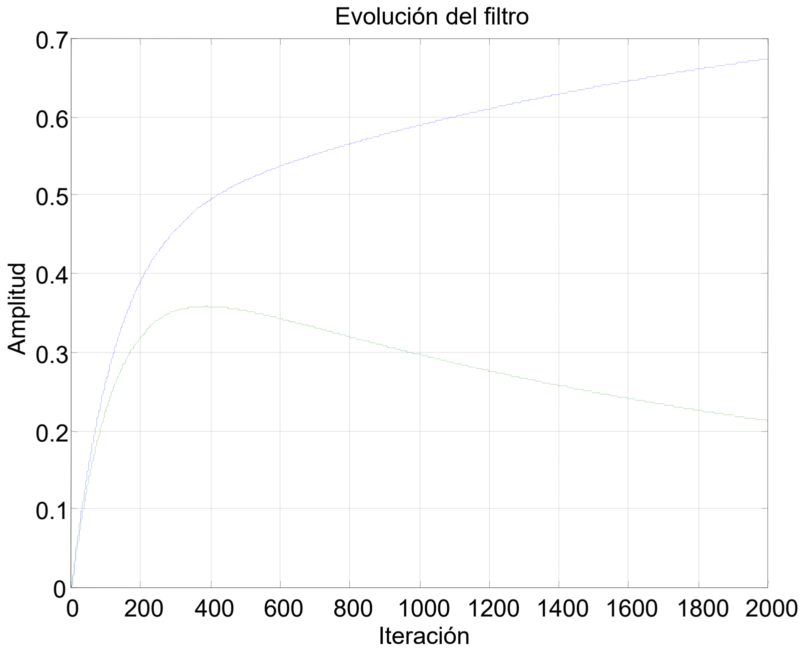


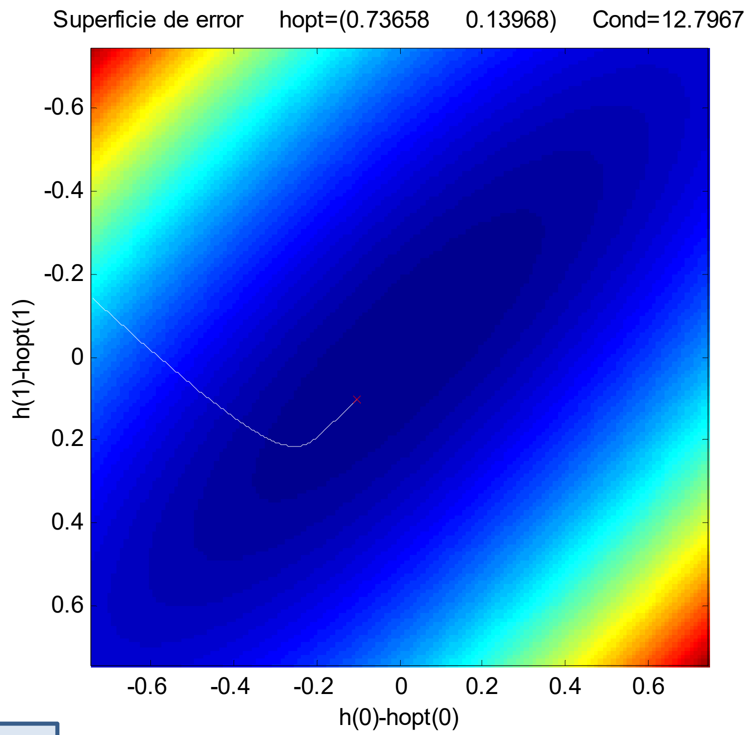
$$\boxed{\mu = 0.005}$$

Convergence & Misadjustment: Examples

Example 2.a: An observation signal ($x[n]$) with **high correlation** between consecutive samples: **high eigenvalue dispersion**.

$$\lambda_{max} = 6.8 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 13 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.75 \quad 0.125]$$



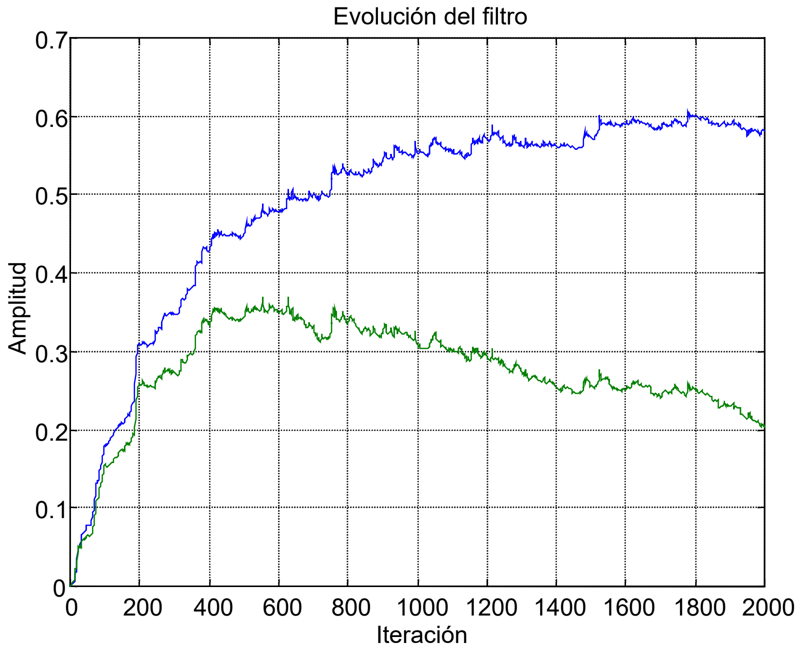


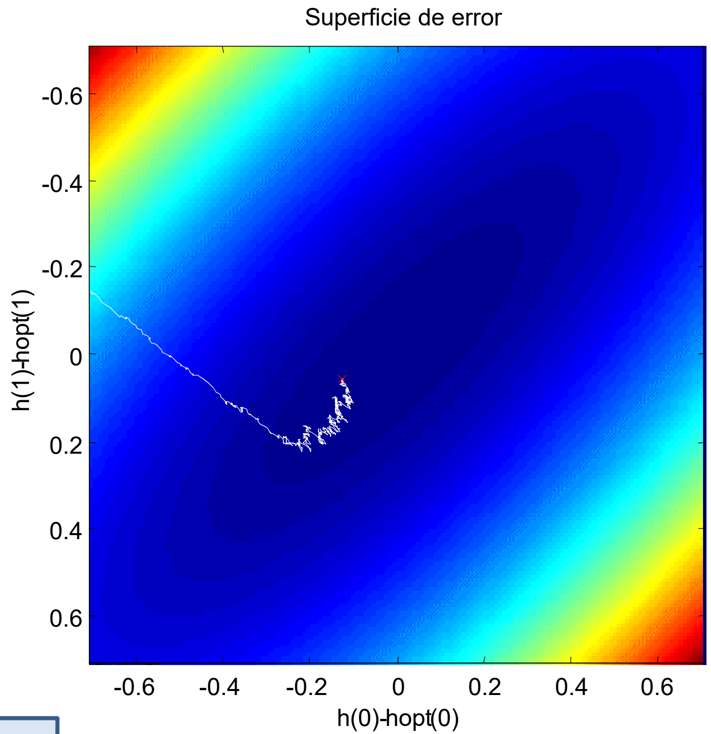
$$\boxed{\mu = 0.001}$$

Convergence & Misadjustment: Examples

Example 2.a: An observation signal ($x[n]$) with **high correlation** between consecutive samples: **high eigenvalue dispersion**.

$$\lambda_{max} = 6.8 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 13 \qquad \underline{\mathbf{h}}_{opt}^T = [0.75 \quad 0.125]$$



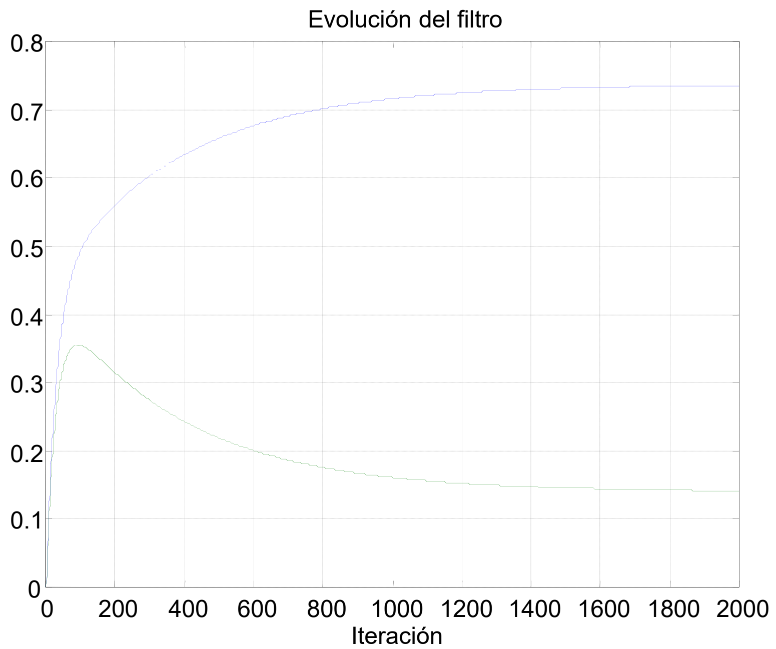


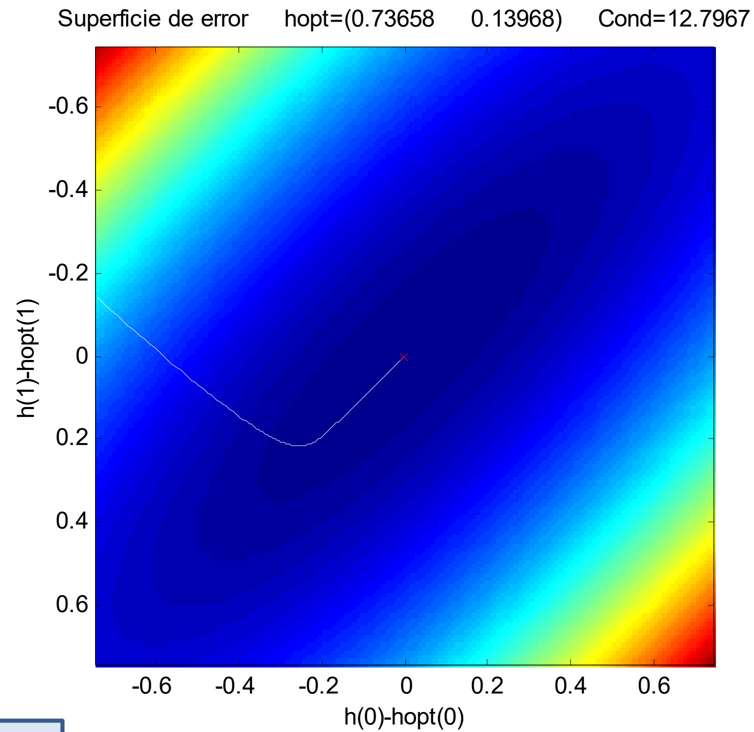
$$\boxed{\mu = 0.001}$$

Convergence & Misadjustment: Examples

Example 2.b: An observation signal ($x[n]$) with **high correlation** between consecutive samples: **high eigenvalue dispersion**.

$$\lambda_{max} = 6.8 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 13 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.75 \quad 0.125]$$





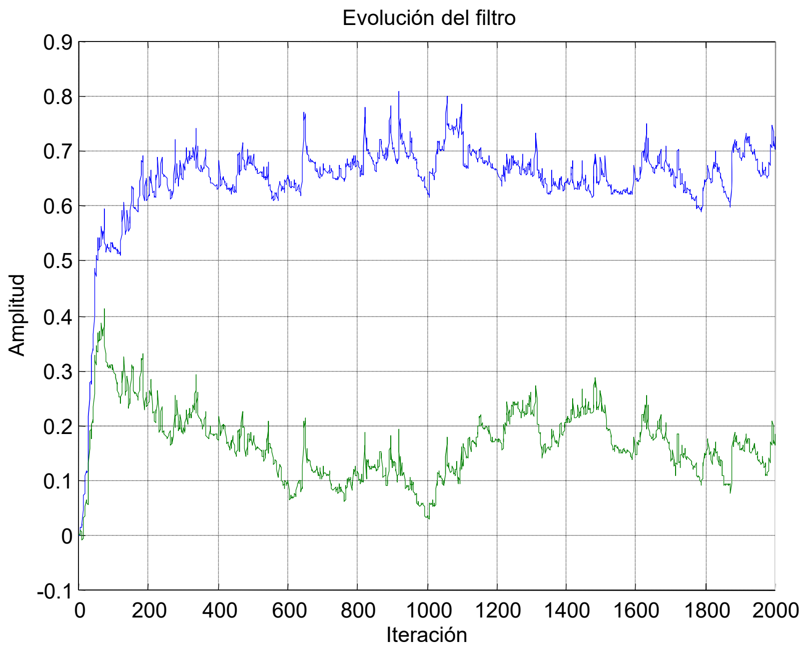
$$\boxed{\mu = 0.005}$$

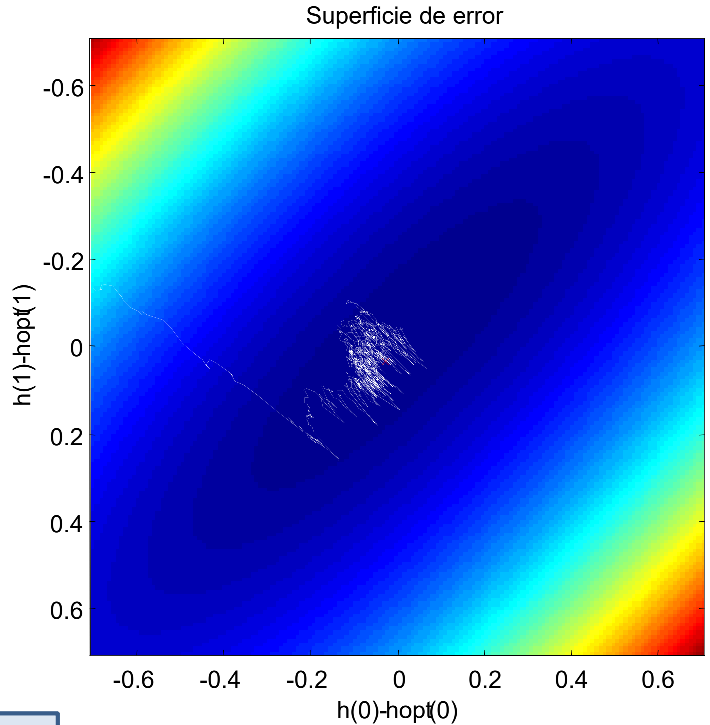
Convergence & Misadjustment: Examples

3.3

Example 2.b: An observation signal ($x[n]$) with **high correlation** between consecutive samples: **high eigenvalue dispersion**.

$$\lambda_{max} = 6.8 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 13 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.75 \quad 0.125]$$





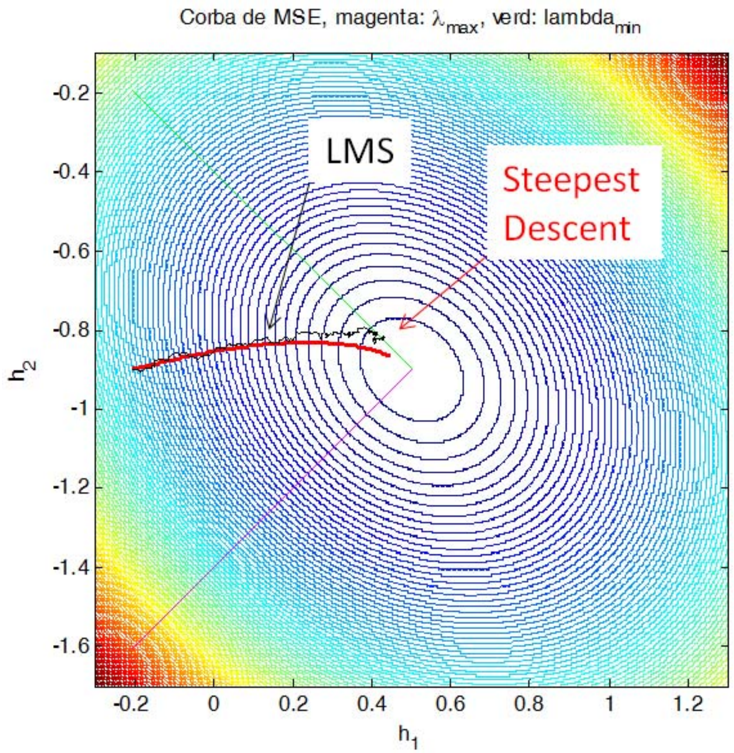
$$\boxed{\mu = 0.005}$$

Convergence & Misadjustment: Examples

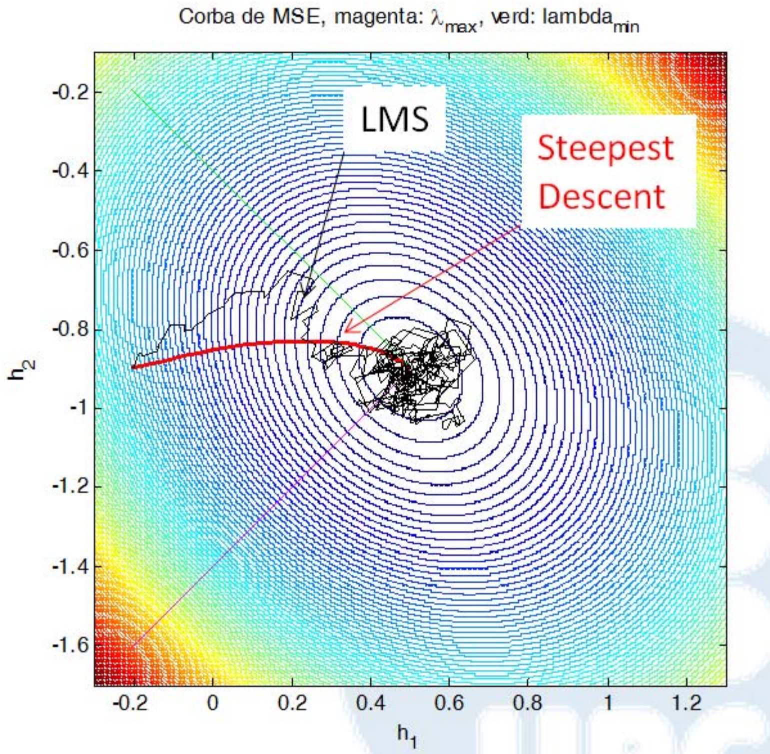
3.3

Example 3.a: Comparison between SD and LSM for an observation ($x[n]$) with **low correlation** between consecutive samples: **low eigenvalue dispersion**.

$$\lambda_{max} = 7.143 \qquad \lambda_{min} = 4.17 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 1.714 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.5 \quad -0.9]$$



$\mu = 0.001$



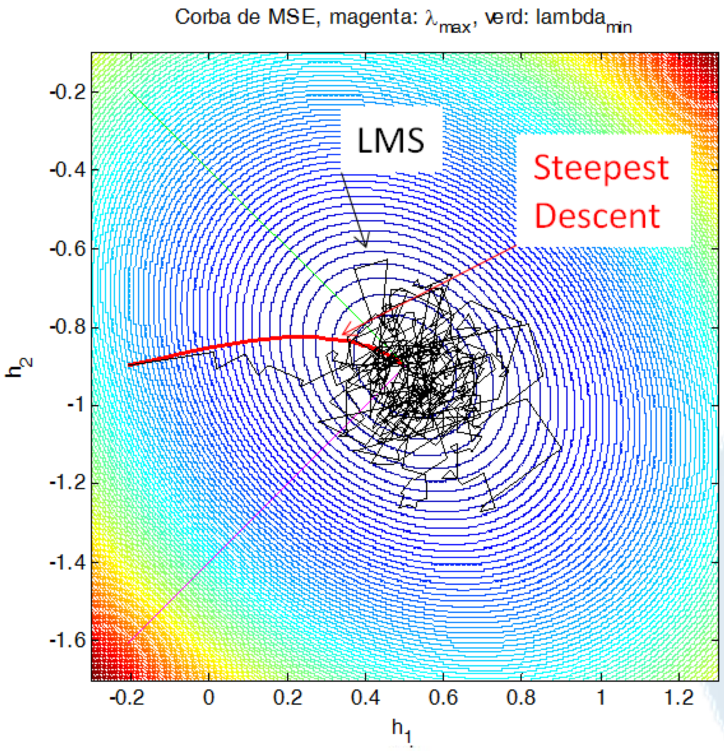
$\mu = 0.01$

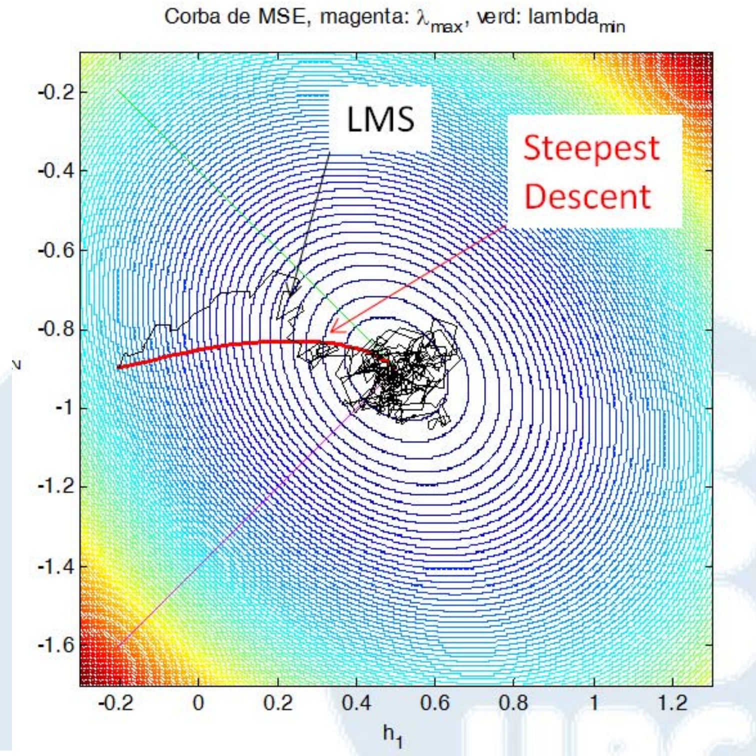
Convergence & Misadjustment: Examples

3.3

Example 3.a: Comparison between SD and LSM for an observation ($x[n]$) with **low correlation** between consecutive samples: **low eigenvalue dispersion**.

$$\lambda_{max} = 7.143 \qquad \lambda_{min} = 4.17 \qquad \frac{\lambda_{max}}{\lambda_{min}} = 1.714 \qquad \underline{\mathbf{h}}_{opt}^{T} = [0.5 \quad -0.9]$$





$\mu = 0.02$ $\qquad$ $\mu = 0.01$

Adaptive Filtering

3.3

1. **Introduction**
    - Scenarios where adaptation is needed
2. **Steepest descend**
    - Study of the error performance surface
    - The minimization algorithm
    - Convergence analysis
3. **Least Mean Square approach**
    - Stochastic approximation of the gradient
    - Convergence analysis
4. **Conclusions**