EE 4745 – Neural Computing

Project 2: Noise Cancellation using Adaptive Filter

Andres Ponce

Javier Solorzano

Donovan Groschang

Electrical Engineering Department

Louisiana State University

15 November 2019

Content

[Introduction 2](#_Toc24451424)

[Mean square error 2](#_Toc24451425)

[1) Eigenvalues and eigenvectors of the Hessian 2](#_Toc24451426)

[2) Minimum point 2](#_Toc24451427)

[3) Contour plot 2](#_Toc24451428)

[LMS algorithm 2](#_Toc24451429)

[4) Maximum stable learning rate 2](#_Toc24451430)

[5) Algorithm implementation 2](#_Toc24451431)

[6) Results 2](#_Toc24451432)

[Noise change 2](#_Toc24451433)

[I. References 3](#_Toc24451434)

[Annex 1: Gantt Diagram 4](#_Toc24451435)

# Introduction

# Mean square error

Following equation 10.12, the mean square error can be written as follow:

## Eigenvalues and eigenvectors of the Hessian

## Minimum point

## Contour plot

# LMS algorithm

## Maximum stable learning rate

## Algorithm implementation

To apply the LMS algorithm to this problem, we’ll consider a bias b = 0 and start with a initial point (or initial weight): . Each following weight will be determined as follow:

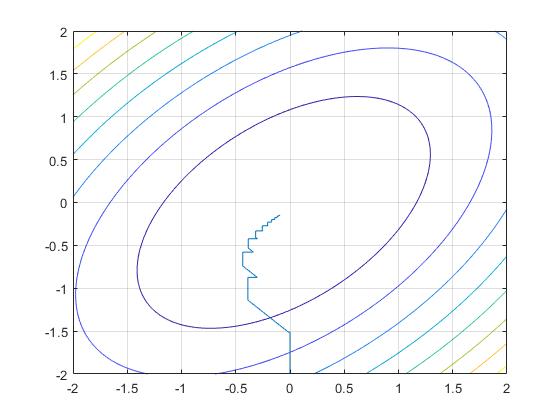
Where:

* e(k) is the difference between the noise added to the signal (actual target) and the output of the adaptive filter:
* is the learning rate; we’ll take as suggested but any value respecting the maximum learning rate condition should allow the algorithm to converge
* is the input vector:

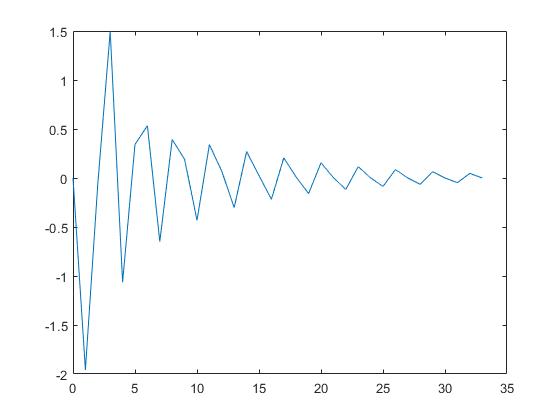
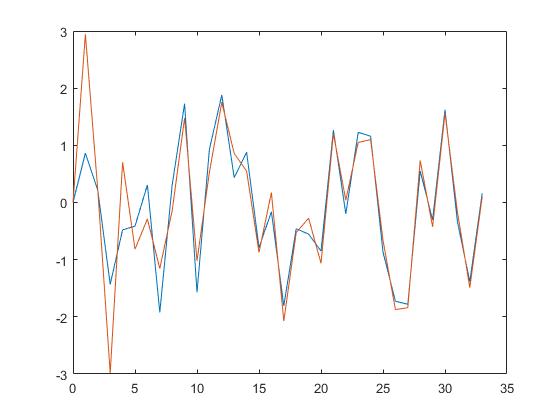
The stopping condition of the algorithm will be: .

The code is attached in Annex 1.

## Results



First, plotting the weight onto the previous contour plot, we can see that the algorithm converges toward a solution which seems close to our minimal point (see Fig. XXXXXXXXX). Then plotting the error, we can see that it is minimized accordingly (see Fig. XXXXXXXXX). Finally, plotting the original signal and the recovered signal, we can see that those two become the same as the error is minimized (see Fig. XXXXXXXXX).



# Noise change

# References

1. “TurtleBot 2 - Open source personal research robot,” *Clearpath Robotics*. [Online]. Available: https://clearpathrobotics.com/turtlebot-2-open-source-robot/. [Accessed: 05-Nov-2019].
2. “Astra Series,” *Orbbec*, 26-Feb-2019. [Online]. Available: https://orbbec3d.com/product-astra-pro/. [Accessed: 05-Nov-2019].
3. “About ROS,” *ROS.org*. [Online]. Available: https://www.ros.org/about-ros/. [Accessed: 05-Nov-2019].
4. “Robot Operating System,” *Wikipedia*, 14-Oct-2019. [Online]. Available: https://en.wikipedia.org/wiki/Robot\_Operating\_System. [Accessed: 05-Nov-2019].
5. “Wiki,” *ros.org*. [Online]. Available: http://wiki.ros.org/Robots/TurtleBot. [Accessed: 05-Nov-2019].
6. V. Mazzari and V. Mazzari, “ROS - Robot Operating System,” *Génération Robots - Blog*, 26-Jul-2016. [Online]. Available: https://www.generationrobots.com/blog/en/ros-robot-operating-system-2/. [Accessed: 05-Nov-2019].