EEL 6504, Spring 2020
Homework 3 QKLMS

March 12, 2020

Due: March 24, 2020

Instructions

Please submit your solutions as a **single PDF file** to the course website on e-Learning at `http://elearning.ufl.edu/`. If you solve any problems by hand, you should scan it as PDF and submit it online.

If your report includes graphics (figures, tables) make sure you have addressed them properly and everything is visible in the final PDF file. Remember commenting your results is very important. **Do your best to clearly address your answers to different parts of each problem**

Your homework submission must cite any references used (including articles, books, code, websites, and personal communications). All solutions must be written in your own words, and you must program the algorithms yourself. If you do work with others, you must list the people you worked with.

If you have any questions address them to the TAs

Problem 1 – 20 points

When a signal is transmitted over a channel, it is distorted because of the noise and the nonlinear characteristic of the channel. The aim of a *channel equalization task*, as shown in Figure.**??**, is to design an inverse model (filter) that represents the best fit to an *unknown noisy and nonlinear plant*, such that when applied to the output of the channel, x_n, it reproduces the original input signal, y_n, as close as possible.

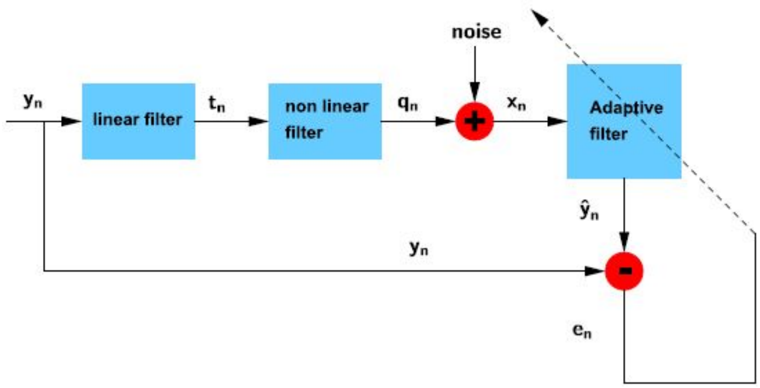

Figure 1: Using adaptive filtering for channel equalization. Reprinted from *Nonlinear Filtering in Reproducing Kernel Hilbert Spaces*, by S. Haykin (You can find this chapter on course webpage).

The purpose of this homework is to demonstrate the performance of the QKLMS in a typical nonlinear channel equalization task. The channel consists of a linear filter

$$t_n = -0.8y_n + 0.7y_{n-1}$$

and a memoryless nonlinearity

$$q_n = t_n + 0.25t_n^2 + 0.11t_n^3$$

The signal q_n is then corrupted by 15dB AWGN and then it is observed as x_n, which is the input of the filter. A delayed version of the plant input y_{n-D} constitutes the desired response.

i. Simulate the dynamics provided by the above equations and generate a data-set of 5000 points. The input signal is Gaussian random variable with zero mean and unit variance.

ii. Set the time delay $D = 2$ and the filter length $l = 5$. Implement the standard LMS, KLMS, and QKLMS for this task. Find the best kernel size for the kernel methods.

iii. Compare the performance of linear and nonlinear filters and show how QKLMS is helpful in this task! Compare the *growth curve* (growth of network size vs iteration no.) of kernel methods. Report and compare the running time of the filters.